

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Радіотехнічний факультет

Кафедра радіотехнічних пристроїв та систем

«На правах рукопису»
УДК 004.932

До захисту допущено:
Завідувач кафедри
_____ Сергій ЖУК
«__» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою

«Радіотехнічні інформаційні технології»

зі спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: «Алгоритм виявлення та оцінювання параметрів руху
маневруючого точкового об'єкту за даними системи
відеоспостереження»**

Виконав (-ла):
студент (-ка) II курсу, групи РТ-91мп
Маленчик Тарас Володимирович _____

Керівник:
Ст.викладач, к.т.н.
Неуймін Олександр Станіславович _____

Рецензент:
Ст.викладач,
Адаменко Володимир Олексійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студент (-ка) _____

Київ – 2020 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Радіотехнічний факультет

Кафедра радіотехнічних пристроїв та систем

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Радіотехнічні інформаційні технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій ЖУК

« 2 » вересня 2020р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Маленчику Тарасу Володимировичу

1. Тема дисертації « Алгоритм виявлення та оцінювання параметрів руху маневруючого точкового об'єкту за даними системи відеоспостереження _____

_____»,
науковий керівник дисертації Неуймін Олександр Станіславович, к.т.н.,
ст.викладач.,

затверджені наказом по університету від « 05 » листопада 2020 р.
№ 3223-с

2. Термін подання студентом дисертації 18 грудня 2020 р.

3. Об'єкт дослідження виявлення та оцінювання параметрів руху маневруючого точкового об'єкту за даними системи відеоспостереження _____

4. Вихідні дані _____
стабільне супроводження маневруючих цілей з прискоренням 30 пк/с^2 ,
виявлення цілей на складному фоні(ВСП 20 дБ) _____

5. Перелік завдань, які потрібно розробити _____
-Провести аналіз методів та алгоритмів виявлення та супроводження точкових об'єктів. _____

- Синтезувати алгоритми виявлення та супроводження точкових об'єктів заданими системи відеоспостереження
 - Провести аналіз синтезованих алгоритмів
 - Розробка стартап-проекту
6. Орієнтовний перелік графічного (ілюстративного) матеріалу _____

7. Орієнтовний перелік публікацій Маленчик Т. В. ОГЛЯД МЕТОДІВ ВІЯВЛЕННЯ ТОЧКОВИХ ЦІЛЕЙ НА ПОСЛІДОВНОСТЯХ ЗОБРАЖЕНЬ / Т. В. Маленчик, О. С. Неуймін. // IX Міжнародна науково-технічна конференція "Радіотехнічні поля, сигнали, апарати та системи" 2020. – 2020.

8. Дата видачі завдання 2 вересня 2020р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Постановка задачі	до 07.09.2020р.	
2	Аналіз та огляд літератури	до 01.10.2020р.	
3	Розробка алгоритмів	до 31.10.2020р.	
4	Статистичний аналіз алгоритмів	до 27.11.2020р.	
5	Розробка стартап-проекту	до 14.12.2020р.	

Студент

Тарас МАЛЕНЧИК

Науковий керівник

Олександр НЕУЙМІН

РЕФЕРАТ

Магістерська дисертація 81с., 32 рис., 14 табл., 26 джерел.

Об'єкт дослідження – виявлення та оцінювання параметрів руху маневруючого точкового об'єкту за даними системи відеоспостереження.

Предмет дослідження – синтез алгоритму виявлення та супроводження маневруючих точкових об'єктів за даними системи відеоспостереження.

Мета роботи – підвищення ефективності виявлення та супроводження точкових об'єктів системою відеоспостереження шляхом синтезу алгоритму виявлення та супроводження маневруючих точкових об'єктів на послідовності відеозображень.

Методи дослідження. У роботі використовувалися: математичний апарат теорії ймовірності, математичної статистики, методи статистичного (імітаційного) моделювання використовуючи ЕОМ і також теорія цифрового оброблення зображень.

Синтезовано алгоритм виявлення та супроводження маневруючої точкової цілі з використанням методів ІРІ та квазіоптимальних фільтрів з зворотніми зв'язками.

Результати дисертації рекомендуються для використання в установах, які займаються створенням і покращенням систем оптичного виявлення БПЛА для підвищення ефективності виявлення і супроводження цілей, що не потребують значних обчислювальних затрат.

Дана робота має важливе значення для підвищення обороноздатності країни. Передбачається що за допомогою синтезованих алгоритмів будуть створені системи для раннього виявлення об'єктів, забезпечення безпеки аеропортів та інших важливих промислових, військових та цивільних об'єктів.

Результати досліджень окремих алгоритмів з дисертації були опубліковані на IX міжнародній науково-технічній конференції "Радіотехнічні поля, сигнали, апарати та системи" 2020.

Ключові слова: виявлення цілі, супроводження цілі, патч-зображення, ІРІ.

ABSTRACT

Master's thesis 81p., 33 figs., 14 tables., 26 sources.

The object of the research is the detection and evaluation of the parameters of the maneuvering point object according to the video surveillance system.

The subject of research is the synthesis of the algorithm for detecting and tracking maneuvering point objects according to the video surveillance system.

The aim of the work is to increase the efficiency of detection and tracking of point objects by a video surveillance system by synthesizing an algorithm for detecting and tracking maneuvering point objects on a sequence of video images.

Research methods. The apparatus of probability theory, mathematical statistics, methods of statistical (simulation) modeling on a computer and the theory of digital image processing were used in the work.

An algorithm for detecting and tracking a maneuvering point target using IPI methods and quasi-optimal feedback filters has been synthesized.

The results of the thesis are recommended for use in institutions involved in the creation and improvement of UAV optical detection systems to increase the efficiency of detection and tracking of targets that do not require significant computational costs.

This work is important for improving the country's defense capabilities. The synthesized algorithms are expected to create systems for early detection, airports security and other important industrial, military and civilian objects.

The research results of certain algorithms were published at the IX International scientific and technical conference "Radioengineering Field, Signals, Devices and Systems" 2020.

Keywords: TARGET DETECTION, TARGET TRACKING, PATCH IMAGE, IPI.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	8
ВСТУП	9
1. ОГЛЯД АЛГОРИТМІВ ВИЯВЛЕННЯ ТОЧКОВОЇ ЦІЛІ	11
2. ОГЛЯД АЛГОРИТМІВ СУПРОВОДЖЕННЯ МАНЕВРУЮЧОЇ ЦІЛІ....	18
2.1 Автономний багатомодельний алгоритм	20
2.2 Узагальнений псевдобаєсівський алгоритм першого порядку	21
2.3 Узагальнений псевдобаєсівський алгоритм другого порядку	21
2.4 Багатомодельний алгоритм з міжмодельною взаємодією.....	23
3. АНАЛІЗ АЛГОРИТМУ ВИЯВЛЕННЯ ТОЧКОВОЇ ЦІЛІ ІРІ.....	25
3.1 Модель патч-зображення	25
3.2 Процедура отримання патч-зображень	25
3.3 Патч-зображення цілі, фону та шуму	27
3.4 Алгоритм виявлення точкової цілі	28
3.5 Короткий опис методу ІРІ	29
3.6 Аналіз методу виявлення точкових цілей ІРІ.....	31
4. СИНТЕЗ АЛГОРИТМУ СУПРОВОДЖЕННЯ МАНЕВРУЮЧОЇ ТОЧКОВОЇ ЦІЛІ ЗА ДАНИМИ ВІДЕОСИСТЕМИ.....	51
4.1 Модель руху об'єкту	51
4.2 Алгоритм супроводження цілі	52
4.3 Аналіз точносних характеристик алгоритму оцінювання параметрів руху цілі з виявленням маневру	54
5. АНАЛІЗ СИНТЕЗОВАНОГО АЛГОРИТМУ ВИЯВЛЕННЯ ТА СУПРОВОДЖЕННЯ МАНЕВРУЮЧОЇ ТОЧКОВОЇ ЦІЛІ	63
6. РОЗРОБКА СТАРТАП ПРОЕКТУ	69
6.1 Ідея продукту	69
6.2 План оновлення продукту.....	70
6.3 Розроблення ринкової стратегії продукту.....	72
6.4 Розроблення маркетингової програми стартап-проекту	74
ВИСНОВКИ.....	77
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	79
ДОДАТКИ.....	82

	7
ДОДАТОК А.....	82
Програма алгоритму виявлення цілі :.....	82
ДОДАТОК Б	86
Програма алгоритму оцінювання параметрів руху цілі з виявленням маневру.....	86

ПЕРЕЛІК СКОРОЧЕНЬ

БПЛА – безпілотні літальні апарати

ВСШ – відношення сигнал/шум

СКВ – середньо квадратичне відхилення

СК – система координат

ЕОМ – електронно-обчислювальна машина

ПЛІС – програмована логічна інтегральна схема

АММ – автономний багатомодельний алгоритм (Autonomous Multiple-Model Algorithm),

GPB1 – узагальнений псевдобаєсівський алгоритм першого порядку (First-Order Generalized PseudoBayesian Algorithm)

GPB2 - узагальнений псевдобаєсівський алгоритм другого порядку (First-Order Generalized PseudoBayesian Algorithm)

ІММ – багатомодельний алгоритм з міжмодельною взаємодією (Interacting Multiple-Model Algorithm)

IPI – infrared patch image

ROI – місця зображення які представляють найбільший інтерес (region of interest)

ВСТУП

Багато країн, щоб забезпечити високий рівень життя та підтримувати макроекономічну стабільність, активно займаються питаннями контролю за територією та забезпечення власної безпеки. Для вирішення цих питань застосовують цифрову обробку зображення, що допомагає швидко, надійно, своєчасно виявляти та відстежувати різні об'єкти. Виявлення об'єктів є базовим етапом для подальшого супроводження цілі. Розроблення нових ефективних методів виявлення та супроводження сприяє покращенню контролю за повітряним простором, запобіганню скоєнню неправомірних дій, створенню систем раннього виявлення тощо. [1]

Перші дослідження раннього виявлення малопомітних та малих за габаритами цілей розпочалися 50-х – 60-х роках XX століття [2]. Цю технологію застосували насамперед в системах спостереження, систем наведення та розвідки [3]. У 80-х роках багато науковців Китаю, США, Франції, Канади почали розробляти та накопичувати теоретичну базу з даного питання. До кінця 1990-х років було розроблено методи, які почали перевірятися та застосовуватися на практиці. В наш час ці методи набули вже достатнього розповсюдження та застосування. Все більше країн світу збільшують нагальність і необхідність даного напрямку, роблять багато інвестицій та наголошують про пріоритетність розвитку методів виявлення малих та нечітких об'єктів на великих відстанях. Також у розвитку цієї галузі зацікавлений і приватний сектор. Станом на сьогодні ведуться роботи для удосконалення і розробки нових методів для швидкого та ефективного виявлення цілей на великих відстанях за різних погодних умов та різних рельєфів.

В наш час безпілотні літальні апарати (БПЛА) набувають великого розвитку і популярності, доступності. Їх використання поширюється на широкий спектр життєдіяльності людини, а саме: у повітряній фото– та відеозйомці, моніторингу природоохоронних об'єктів, забезпечення

рятувально-пошукових операцій, спостереження за радіаційним, біологічним, екологічним та ін. станами, спостереження за територією, охороні стратегічно важливих об'єктів інфраструктури, тощо.

З появою і розвитком БПЛА, з'явився новий клас потенційних небезпек: використання БПЛА для введення незаконної фото- та відеозйомки, застосування у терористичних цілях, проникнення на заборонену територію та проведення там несанкціонованих дій, загроза для повітряного транспорту, загроза польотів у місцевості з щільною забудовою, можливі падіння даних апаратів, що може нанести шкоду оточуючому середовищу,

Отже виникає потреба у розробці нових, сучасних систем, які були б здатні виконувати задачі виявлення і супроводження цілі. Оптична система спостереження та супроводження цілі, може виконувати подібні завдання, так як вона поєднує у собі функції цифрової обробки зображень та радіолокації. Такі системи мають здатність виявляти БПЛА та інші точкові об'єкти в умовах поганої видимості, на великих відстанях, в складних погодних умовах та в на різних природних ландшафтах (гори, море та ін.), а також характеризуються високою якістю зображення, та малими габаритами. Відповідно вони можуть бути корисними у військовій сфері – для раннього виявлення ворожих засобів та ефективної протидії їм, екологічній сфері – виявлення та моніторинг пожеж, охоронній сфері – для стеження та контролю за територією, та ін. [2],

Все вище перераховане вказує на актуальність розробки відповідних алгоритмів виявлення та супроводження для систем та комплексів відеоспостереження, які були б здатні зменшити ризики від діяльності БПЛА.

1. ОГЛЯД АЛГОРИТМІВ ВИЯВЛЕННЯ ТОЧКОВОЇ ЦІЛІ

При вирішенні задачі контролю за повітряним простором важливо щоб ціль була виявлена на максимально великій відстані, де ціль розглядається як точкова, така що може займати лише декілька пікселів на зображенні. Це пов'язано з складною навколишньою обстановкою: атмосферними явищами, нерівномірністю фону, малою помітністю цілі та малим ВСШ. Вище зазначені причини роблять цю задачу не тривіальною і провокують створення нових методів для її вирішення [4].

Усі відомі методи можна умовно поділити на два види - «супроводження до виявлення» (рис.1.1) та «виявлення до супроводження» (рис.1.2) [5, 6]. Методи, які використовують «супроводження до виявлення», визначають траєкторію цілі, після цього вже виноситься рішення про виявлення цілі, за результатами обробки декількох кадрів. Цей спосіб використовує просторову та часову інформацію, використовуючи спеціально розроблені процедури часової та просторово – часової фільтрації. Якщо задана та реальна швидкості об'єктів не співпадають, або ціль здійснює маневрування, алгоритми починають втрачати свою початкову ефективність.

Також відомі методи динамічного програмування та методи з використанням перетворень Хафа і Родона, ієрархічний метод виявлення, який ґрунтується на використанні технології відсіваючого експерименту методом Хока. Такі підходи усувають недолік неузгодженості швидкостей та дозволяють супроводжувати та виявляти слабоманевруючі об'єкти [4].

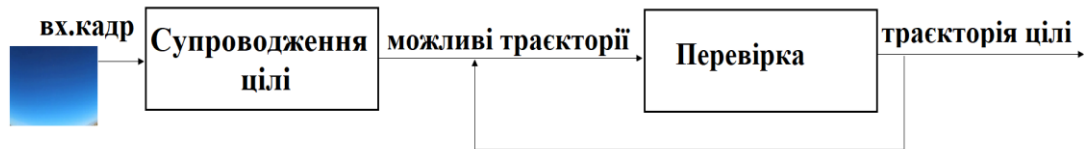


Рисунок.1.1 Модель виявлення об'єкта методом «супроводження до виявлення»

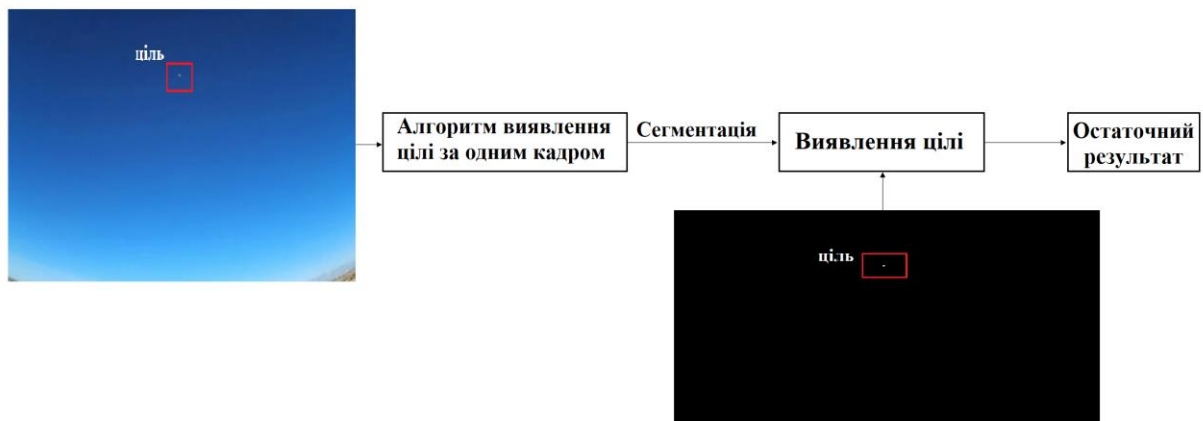


Рисунок.1.2 Модель виявлення об'єкта методом «виявлення до супроводження»

До інших недоліків можна віднести так звану “пакетність”, для роботи цих алгоритмів потрібно накопичити деяку кількість кадрів, що в свою чергу веде до появи певної затримки, при появі результатів. Так як ці методи є безпороговими, це призводить до використання великої кількості даних для обробки [4].

Методи, які використовують принцип «виявлення до супроводження» не потребують процедури накопичення кадрів чи знання про швидкість, траєкторію чи форму цілі [6]. Рішення про виявлення цілі приймається у кожному кадрі. Подальші ж результати передаються на алгоритм супроводження за об'єктом.

В загальному вигляді модель зображення з системи відеоспостереження складається з цілі, фону і шуму:

$$f_0(x, y) = f_T(x, y) + f_B(x, y) + f_N(x, y), \quad (1.1)$$

де x та y координати пікселів. Відштовхуючись від цього, можна сказати, що зі збільшенням відстані показники шуму зростають і ціль втрачає свої контури, стає розмитою, через це її розміри на зображенні можуть бути зменшені аж до декількох пікселів [6]. Тому для представлення цілі, використовують двовимірну гаусівську модель:

$$I_T(x, y) = I_{\max} \exp\left(-\frac{1}{2} \left[\left(\frac{x}{\sigma_x}\right)^2 + \left(\frac{y}{\sigma_y}\right)^2\right]\right), \quad (1.2)$$

де ціль представлена величиною піковою інтенсивності I_{\max} , а σ_x , σ_y - параметри горизонтальної та вертикальної протяжності.

На основі такої моделі було запропоновано чимало алгоритмів, таких як:

- методи виявлення по одному кадру,
- методи, що базуються на помітності цілі,
- методи виявлення цілі на основі патч-рівня,
- методи виявлення цілі на основі рівня патч-зображення,
- методи засновані на вейвлет перетворенні,
- методи засновані на міжкадровій різниці.

Математичний апарат з застосуванням **вейвлет функцій** широко застосовується у цифровій обробці зображення [7-12]. Ці алгоритми мають чимало переваг таких як: здатність повністю відновити зображення, структурувати зображення, тобто виділити місця зображення, які є найбільш цікавими для подальшого вивчення (region of interest - ROI), високі обчислювальні можливості та здатність до аналізу багато-роздільним методом, що дозволяє аналізувати зображення у різних площинах [13]. Це дозволяє відділити фон зображення від ROI. Недоліком цих алгоритм є їх низька ефективність при низькому ВСШ.

Алгоритми, які застосовують **метод міжкадрової різниці** ґрунтуються на визначенні різниці сірого між двома послідовними кадрами, що може допомогти у оцінці руху об'єкта. Щоб визначити цю різницю

встановлюють границю T , якщо різниця менше встановленого порогу, то цілі немає і навпаки [13]. Недоліками даного підходу являється емпіричний вибір порогу T . Існує багато варіантів вибору його встановлення, таких як: встановлення для кожного кадру окремо, для всієї послідовності та ін. Все це призводить до не завжди ефективної роботи алгоритмів. До інших недоліків варто віднести низьку ефективність при його використанні на складному фоні, який не контрастує з ціллю [13]. Для покращення цього алгоритму застосовують додатковий математичний апарат (самоадаптивні алгоритми [14], алгоритми засновані на статистиці вищих порядків [15] та ін), що погіршує простоту, а звідси і швидкість алгоритму. Для збільшення вірогідності виявлення цілі використовують статистичні моделі та перевірки різноманітних гіпотез. Також ці алгоритми можуть помилятися у виявленні цілі (так звана хибна тривога). Основними проблемами цих алгоритмів є складність супроводження об'єкта та подальша обробка з метою розпізнання реальної, а не хибної цілі. Отже для покращення ефективності даних алгоритмів, необхідно доповнювати їх функціонал іншими методами.

Методи виявлення за одиночним кадром ґрунтуються на дослідженні та прогнозуванні фону зображення, з припущенням що фон майже не змінюється та має велику кореляцію, на відміну від цілі. Такі методи придушують фонові завади, за рахунок чого покращують видимість цілі.

Для того щоб не сплутати фон зображення та шуми, які його оточують, використовують спеціальні фільтри, за допомогою яких можна оцінити фон зображення, і після чого виділити ціль. Це робиться завдяки відніманню фону зображення від оригінального кадру.

$$f_T(x, y, k) = f(x, y, k) - f_B(x, y, k), \quad (1.3)$$

До недоліків даного підходу можна віднести те, що під час процесу фільтрації зображення, ціль може бути відфільтрована разом з фоном чи супутніми шумами. До цих методів відносять такі класичні підходи як: Two

dimensional least mean square (TDLMS), TopHat, MaxMean та MaxMedian. TDLMS фільтри є адаптивними, і призначені для оцінки майбутньої зміни фону зображення. Для успішної роботи даних фільтрів використовується велика кількість параметрів:

- вибір кроку (лінійний, нелінійний, емпіричний, тощо),
- вибір ROI,
- вибір вікна, та ін.

Фільтри MaxMean та MaxMedian [16], які базуються на характеристиці розподілу сірого навколо цілі, здатні виявити об'єкт шляхом подавлення завад фону, але є дуже чутливим до зміни власних параметрів. Для боротьби з цим недоліками були розроблені фільтри [17], які є менш чутливими до параметрів, що впливають на виявлення цілі, проте не вдається досягти гарних результатів на зашумленому фоні з малим ВСШ. Головним ж недоліком подібного підходу є низька спроможність працювати з дуже малими (точковими) об'єктами та неспроможність працювати з зображеннями, які не мають контрастного фону, що ускладнює виявлення цілі.

Підсумовуючи роботу методів з використанням фільтрів можна відміти, що вони не можуть повністю прибрати шум з зображення, а лише придушують його [18] (зокрема Median фільтр не справляється з гаусівським шумом), а деякі з них мають велику обчислювальну складність, що зменшує швидкодію алгоритму. Тож дані алгоритми слід використовувати з іншими, щоб забезпечити необхідний рівень виявлення точкового об'єкту.

Методи, що базуються на помітності цілі використовують властивості людського ока (бінокулярність, розрізнення контрасту, яскравості), яке звертає свою увагу на ціль, на зображенні [19]. Відповідно до цих особливостей, були розроблені методи просторової фільтрації, методи з використанням карт помітності цілі (використовується здатність людини розрізнити контраст), методи на основі фазового спектру. Ці

алгоритми також застосовують спеціальні фільтри для оцінки фону зображення. Дані алгоритми демонструють свої переваги на зображеннях, на яких фон легко відрізнити від об'єкта, за яким ведеться спостереження. При роботі у складних умовах, дані методи вже не так ефективно виконують свою задачу. Швидкодія роботи алгоритму падає, а звідси і його ефективність. Для покращення їх роботи необхідно налаштовувати фільтри, та володіти знанням про параметри цілі, що виявляється [13]. Ще одним суттєвим недоліком таких алгоритмів є нездатність працювати з об'єктами малого розміру, з нечіткими границями.

Методи виявлення цілі на основі патч-рівня використовують шаблони для розпізнавання. Зображення розбивається на менші частини - патчі, далі розпізнаються фон та ціль. Цей підхід схожий на той, що використовується у задачі розпізнавання обличчя. Дані алгоритми використовують статистичний математичний апарат – метод головних компонент. Використання шаблонів для розпізнавання робить цей метод більш складним та громіздким. Для його ефективної роботи використовують декілька наборів шаблонів для правильного визначення чи патч відноситься до зображення фону чи до зображення цілі [6]. Обчислювальна складність не дозволяє цим методам швидко та ефективно виявляти ціль.

Методи виявлення цілі на основі рівня патч-зображення використовують властивість високої кореляції несусідніх патчів фону. Зображення ділиться на окремі патчі за допомогою ковзного вікна, з отриманих патчів формується вектор нового зображення – патч-зображення. [4]. Далі проводиться оцінювання фону та цілі, після чого проводиться реконструкція зображення з використанням патчів [6]. Для найбільш ефективного використання таких алгоритмів, слід правильно обирати розміри ковзного вікна та його крок. Такі методи добре зарекомендували себе в роботі з малими об'єктами, з нерівномірною яскравістю, які знаходяться на зашумленому фоні. Ці алгоритми мають

менше залишкових шумів і завад, в порівнянні з іншими методами, що дає змогу зменшити вірогідність помилки. До недоліків методів виявлення цілі на основі рівня патч-зображення відносять його обчислювальну складність.

Таблиця 1.1 Порівняння методів виявлення

Метод	Переваги	Недоліки
Методи виявлення по одному кадру	Прості в реалізації	Неефективні при складному фоні. Необхідність фільтрів, які відповідають параметрам цілі
Методи, що базуються на помітності цілі	Прості в реалізації	Неефективні для дуже малих та непомітних об'єктів
Методи виявлення на основі патч-рівня	Ефективно вирізняють ціль на складному зашумленому фоні	Використання словників для розпізнавання
Методи виявлення на основі рівня патч-зображення	Ефективно виявляють ціль на складному фоні	Обчислювальна складність

З таблиці 1.1 видно, що методи виявлення цілі на основі рівня патч – зображення найкраще справляються з завданням виявлення цілі на складному фоні. Тому варто обрати саме такий метод, для подальшого його покращення та нівелювання його головного недоліка, шляхом оптимального вибору його параметрів.

2. ОГЛЯД АЛГОРИТМІВ СУПРОВОДЖЕННЯ МАНЕВРУЮЧОЇ ЦІЛІ

Важливим завданням оброблення координатної інформації про рух об'єктів, що маневрують, є фільтрація параметрів їх руху. Коли відомі параметри маневру об'єкта, тоді для вирішення цього завдання застосовують одномодельний дискретний фільтр Калмана [20]. На практиці ж, ці параметри часто не відомі, тому такі об'єкти описують застосовуючи дискретні динамічні системи, параметри чи структура яких відомі з точністю до набору гіпотез, і змінюються стрибкоподібно в випадкові моменти часу. Використання баєсівського метода оцінювання процесів з випадковою структурою в дискретному часі веде до використання оптимальних фільтрів з зростаючим числом каналів, і не дозволяє отримати рекурентні оптимальні алгоритми оцінювання стану та управління динамічною системою з випадковою структурою. Це обумовлено тим, що він не враховує Марківську властивість розширеного змішаного процесу, що включає неперервний процес з випадковою структурою і дискретний компонент, що характеризує вид його структури. Тому для оцінювання стану і управління динамічною системою з випадковою структурою доцільно застосувати апарат змішаних марківських процесів в дискретному часі.

Нелінійну маневрену поведінку цілі можна уявити безліччю лінійних моделей. У кожен момент часу k поведінка цілі характеризується однією з цих моделей. Зміна моделей є випадковим марківським процесом з певною ймовірністю переходу з одного стану в інший. Відповідно алгоритми фільтрації, що використовують такий підхід, вважаються *багатомодельними*.

Багатомодельний алгоритм фільтрації складається з таких операцій [21]:

- переініціалізація фільтрів, для забезпечення взаємодії між моделями,
- розрахунок оцінок $\hat{x}_k^{(i)}$ та їх коваріацій $\hat{P}_k^{(i)}$ для кожного з N – фільтрів, налаштованих на відповідну модель,
- об'єднання оцінок окремих фільтрів в загальну \hat{x}_k ,
- пошук об'єднаної коваріації \hat{P}_k , враховуючи вагові коефіцієнти \hat{W}_k .

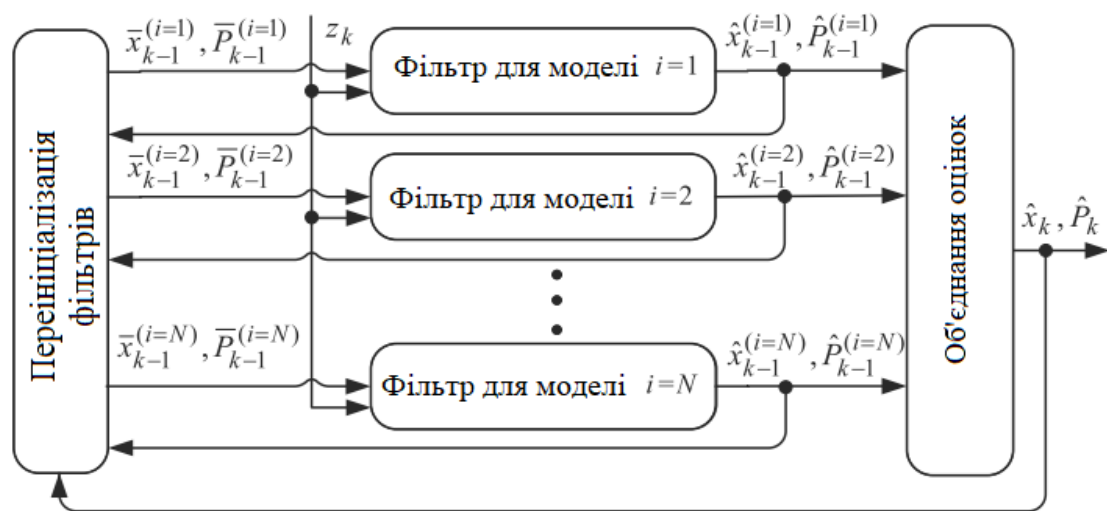


Рисунок. 2.1 Структурна схема багатомодельного алгоритму

Найбільш розповсюдженими алгоритмами являються:

- автономний багатомодельний алгоритм (Autonomous Multiple-Model Algorithm - AMM),
- узагальнений псевдобаєсівський алгоритм першого порядку (First-Order Generalized PseudoBayesian Algorithm - GPB1),
- узагальнений псевдобаєсівський алгоритм другого порядку (First-Order Generalized PseudoBayesian Algorithm - GPB2),
- багатомодельний алгоритм з міжмодельною взаємодією (Interacting Multiple-Model Algorithm - IMM).

2.1 Автономний багатомодельний алгоритм

Такий алгоритм складається з N дискретних калманівських фільтрів, налаштованих на різні математичні моделі, що працюють паралельно. Переініціалізація фільтрів не виконується, а оцінки кожного з фільтрів є незалежними від оцінок інших фільтрів.

Алгоритм виконується по наступним крокам:

- 1) вираховуються оцінки фільтра $\hat{x}_k^{(i)}$ та їх коваріації $\hat{P}_k^{(i)}$.

$$\tilde{x}_k^{(i)} = F_{k-1}^{(i)} \tilde{x}_{k-1}^{(i)}; \tilde{P}_k^{(i)} = F_{k-1}^{(i)} \tilde{P}_{k-1}^{(i)} (F_{k-1}^{(i)})^T + Q_{k-1}^{(i)}; \quad (2.1)$$

$$\hat{x}_k^{(i)} = \tilde{x}_k^{(i)} K_k^{(i)} v_k^{(i)}; \hat{P}_k^{(i)} = (I - K_k^{(i)} H_k^{(i)}) \tilde{P}_{k-1}^{(i)}; \quad (2.2)$$

$$K_k^{(i)} = \tilde{P}_k^{(i)} (H_k^{(i)})^T (V_k^{(i)})^{-1}; V_k^{(i)} = H_k^{(i)} \tilde{P}_k^{(i)} (H_k^{(i)})^T + R_k^{(i)}, \quad (2.3)$$

$$v_k^{(i)} = z_k - H_k^{(i)} \tilde{x}_k^{(i)},$$

де I – одинична матриця, $K_k^{(i)}$ – матриця коефіцієнтів підсилення фільтру Калмана для i – моделі в момент часу k , $v_k^{(i)}$, $V_k^{(i)}$ – нев'язка та її коваріація для i -ї моделі, $H_k^{(i)}$, $F_k^{(i)}$ – матриці переходу та вимірювання, z_k – вектор вимірювань. $R_k^{(i)}$, $Q_{k-1}^{(i)}$ – коваріаційні матриці некорельованих між собою гаусівських випадкових послідовностей.

- 2) рахуються вагові коефіцієнти фільтру $\hat{W}_k^{(i)}$.

$$\hat{W}_k^{(i)} = \frac{p(z_k | s_k = i) \tilde{W}_k^{(i)}}{\sum_{j=1}^N p(z_k | s_k = j) \tilde{W}_k^{(j)}}; \hat{W}_k^{(i)} = \sum_{j=1}^N q^{(ji)} \hat{W}_{k-1}^{(j)}, \quad (2.4)$$

$$p(z_k | s_k = i) = p(v_k^{(i)}) = \mathcal{N}[v_k^{(i)} | 0, V_k^{(i)}]; \quad (2.5)$$

$$\mathcal{N}[\alpha | m, \Sigma] = (2\pi)^{-r/2} \det(\Sigma)^{-1/2} \exp\{-\frac{1}{2}(\alpha - m)^T \Sigma^{-1}(\alpha - m)\}, \quad (2.6)$$

де $\mathcal{N}[\alpha | m, \Sigma]$ – гаусівська щільність ймовірності r -мірної випадкової величини α з мат. очікуванням m і дисперсією Σ , s_k – індекс структури, $q^{(ji)}$ – ймовірність переходу з одного стану в інший.

- 3) виконується об'єднання оцінок окремих фільтрів в загальну оцінку \hat{x}_k і знаходження коваріації \hat{P}_k :

$$\hat{x}_k = \sum_{i=1}^N \hat{W}_k^{(i)} \hat{x}_k^{(i)}; \hat{P}_k = \sum_{i=1}^N [\hat{P}_k^{(i)} + (\hat{x}_k - \hat{x}_k^{(i)})(\hat{x}_k - \hat{x}_k^{(i)})^T] \hat{W}_k^{(i)} \quad (2.7)$$

2.2 Узагальнений псевдобаєсівський алгоритм першого порядку

Оцінки фільтрів попередніх кроків $\hat{x}_{k-1}^{(i)}$ та їх коваріації $\hat{P}_{k-1}^{(i)}$ на поточній ітерації замінюються підсумковою оцінкою \hat{x}_{k-1} та відповідною коваріацією \hat{P}_{k-1} .

Алгоритм виконується по наступним крокам:

- 1) відбувається переініціалізація фільтрів (рис 2.2):

$$\bar{x}_{k-1}^{(i)} = \hat{x}_{k-1}^{(i)}; \bar{P}_{k-1}^{(i)} = \hat{P}_{k-1}^{(i)}, \quad (2.8)$$

- 2) вираховуються оцінки фільтра $\hat{x}_k^{(i)}$ та їх коваріації $\hat{P}_k^{(i)}$ (2.1 – 2.3),
- 3) рахуються вагові коефіцієнти фільтру $\hat{W}_k^{(i)}$ (2.4 – 2.6),
- 4) виконується об'єднання оцінок окремих фільтрів в загальну оцінку \hat{x}_k і знаходження коваріації \hat{P}_k (2.7).

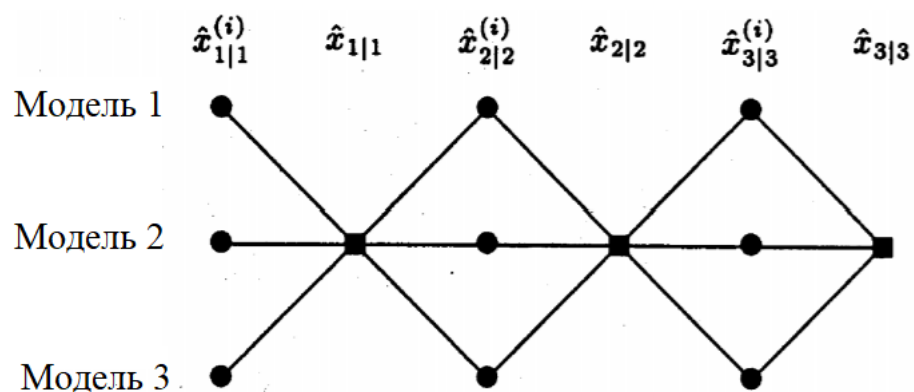


Рисунок. 2.2 Переініціалізація узагальненого псевдобаєсівського алгоритму першого порядку

2.3 Узагальнений псевдобаєсівський алгоритм другого порядку

Даний алгоритм містить N^2 калманівських фільтрів. Визначення оцінки відбувається по кожній можливій моделі на поточній та минулій ітераціях, сумарно виходить N^2 гіпотез.

Алгоритм виконується за наступними кроками:

1) переініціалізація фільтрів (рис. 2.3):

$$\bar{x}_{k-1}^{(ij)} = \hat{x}_{k-1}^{(ij)}; \bar{P}_{k-1}^{(ij)} = \hat{P}_{k-1}^{(ij)}, \quad (2.9)$$

2) вираховуються оцінки фільтра $\hat{x}_k^{(ij)}$ та їх коваріації $\hat{P}_k^{(ij)}$:

$$\tilde{x}_k^{(ij)} = F_{k-1}^{(i)} \bar{x}_{k-1}^{(ij)}; \tilde{P}_k^{(ij)} = F_{k-1}^{(i)} \bar{P}_{k-1}^{(ij)} (F_{k-1}^{(i)})^T + Q_{k-1}^{(i)}; \quad (2.10)$$

$$\hat{x}_k^{(ij)} = \tilde{x}_k^{(ij)} K_k^{(ij)} v_k^{(ij)}; \hat{P}_k^{(ij)} = (I - K_k^{(ij)} H_k^{(i)}) \tilde{P}_k^{(ij)}; \quad (2.11)$$

$$K_k^{(ij)} = \tilde{P}_k^{(ij)} (H_k^{(i)})^T (V_k^{(ij)})^{-1}; V_k^{(ij)} = H_k^{(i)} \tilde{P}_k^{(ij)} (H_k^{(i)})^T + R_k^{(i)},$$

$$v_k^{(ij)} = z_k - H_k^{(i)} \tilde{x}_k^{(ij)}, \quad (2.12)$$

3) рахуються вагові коефіцієнти фільтра $\hat{W}_k^{(i)}$, сумісних оцінок та їх коваріацій:

$$\bar{W}_k^{(j|i)} = \frac{\mathcal{N}[v_k^{(ji)} | 0, V_k^{(ji)}] q^{(ji)} \hat{W}_{k-1}^{(j)}}{\bar{W}_k^{(i)}}; \tilde{W}_k^{(i)} = \sum_{j=1}^N \mathcal{N}[v_k^{(ji)} | 0, V_k^{(ji)}] q^{(ji)} \hat{W}_{k-1}^{(j)}, \quad (2.13)$$

$$\hat{x}_k^{(i)} = \sum_{j=1}^N \tilde{W}_k^{(j|i)} \hat{x}_k^{(ji)}; \hat{P}_k^{(i)} = \sum_{j=1}^N [\hat{P}_k^{(ji)} + (\hat{x}_k^{(i)} - \hat{x}_k^{(ji)})(\hat{x}_k^{(i)} - \hat{x}_k^{(ji)})^T] \tilde{W}_k^{(j|i)} \quad (2.14)$$

4) виконується об'єднання оцінок окремих фільтрів в загальну оцінку \hat{x}_k і знаходження коваріації \hat{P}_k :

$$\hat{W}_k^{(i)} = \frac{\tilde{W}_k^{(i)}}{\sum_{j=1}^N \tilde{W}_k^{(j)}}; \quad (2.15)$$

$$\hat{x}_k = \sum_{j=1}^N \hat{W}_k^{(j)} \hat{x}_k^{(j)}; \hat{P}_k = \sum_{j=1}^N [\hat{P}_k^{(j)} + (\hat{x}_k - \hat{x}_k^{(j)})(\hat{x}_k - \hat{x}_k^{(j)})^T] \hat{W}_k^{(j)} \quad (2.16)$$

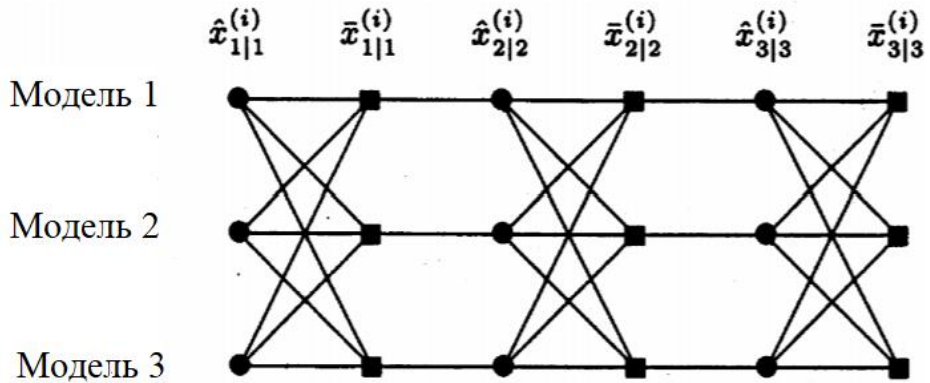


Рисунок. 2.3 Переініціалізація узагальненого псевдобасейського алгоритму другого порядку

2.4 Багатомодельний алгоритм з міжмодельною взаємодією

Такий алгоритм складається з N дискретних калманівських фільтрів, які переініціалізуються з врахуванням оцінок минулих фільтрів (рис.2.4).

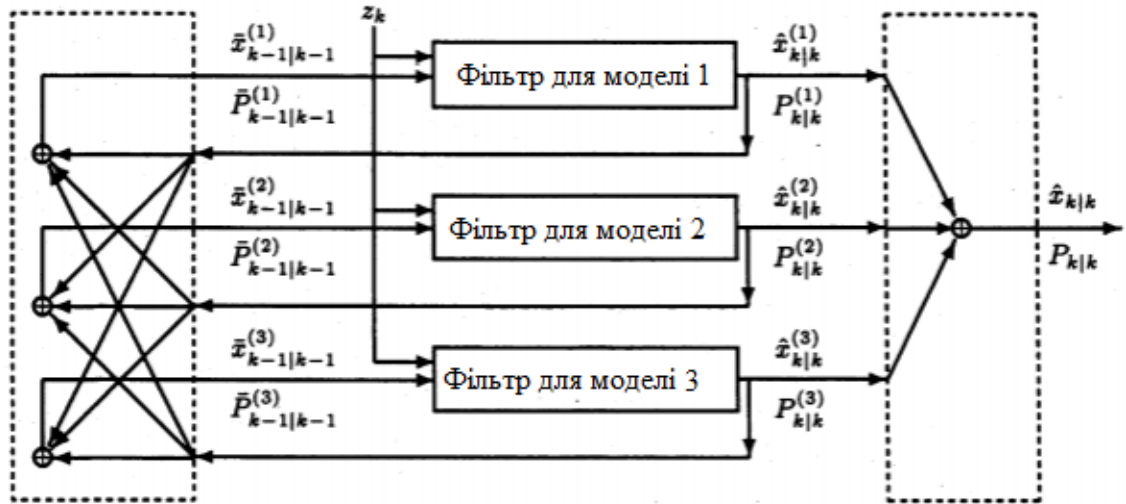


Рисунок. 2.4 Структурна схема багатомодельного алгоритму з міжмодельною взаємодією

Алгоритм виконується по наступним крокам:

1) переініціалізація фільтрів (рис. 2.5):

$$\tilde{W}_k^{(i)} = \sum_{j=1}^N q^{(ji)} \hat{W}_{k-1}^{(j)}; \quad \hat{W}_{k-1}^{(j|i)} = \frac{q^{(ji)} \hat{W}_{k-1}^{(j)}}{\tilde{W}_k^{(i)}}, \quad (2.17)$$

$$\bar{x}_{k-1}^{(i)} = \sum_{j=1}^N \hat{x}_{k-1}^{(j)} \hat{W}_{k-1}^{(j|i)}; \quad \bar{P}_{k-1}^{(i)} = \sum_{j=1}^N [\hat{P}_{k-1}^{(j)} + (\hat{x}_{k-1}^{(j)} - \hat{x}_{k-1}^{(i)})(\hat{x}_{k-1}^{(i)} - \hat{x}_{k-1}^{(j)})^T] \hat{W}_{k-1}^{(j|i)}; \quad (2.18)$$

- 2) вираховуються оцінки фільтра $\hat{x}_k^{(i)}$ та їх коваріації $\hat{P}_k^{(i)}$ (2.1 – 2.3),
- 3) рахуються вагові коефіцієнти фільтра $\hat{W}_k^{(i)}$ (2.4 – 2.6),
- 4) виконується об'єднання оцінок окремих фільтрів в загальну оцінку \hat{x}_k і знаходження коваріації \hat{P}_k (2.7).

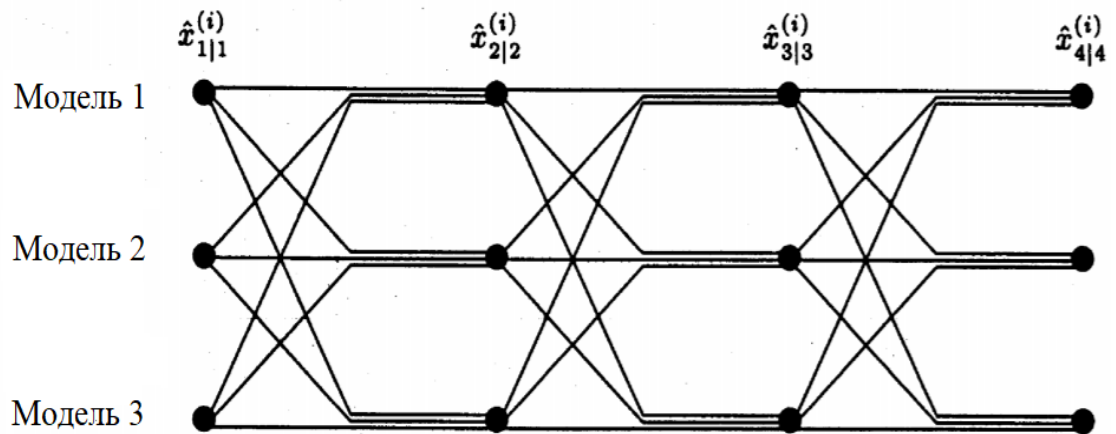


Рисунок. 2.5 Переініціалізація багатомодельного алгоритму з міжмодельною взаємодією

Враховуючи вищеперераховані алгоритми, метод АММ вимагає найменших обчислювальних затрат та є простим у реалізації, однак найкращим з точки зору «точність – обчислювальні затрати» є алгоритм ІММ [21].

Отже, для оцінювання параметрів руху маневруючого об'єкту доцільно застосувати апарат змішаних марківських процесів в дискретному часі. В подальшій частині роботи пропонується синтезувати алгоритм оцінювання параметрів руху маневруючого об'єкту за даними системи відеоспостереження за методикою, що представлена в [20], оскільки вона частково повторює кроки алгоритму ІММ.

3. АНАЛІЗ АЛГОРИТМУ ВИЯВЛЕННЯ ТОЧКОВОЇ ЦІЛІ ІРІ

Серед методів виявлення цілі на основі рівня патч-зображення найкращими показниками ефективності виявлення цілі володіє метод на основі моделі інфрачервоного патч-зображення ІРІ [22]. Метод дозволяє отримати менше залишкових шумів і завад для різноманітних фонів, в порівнянні з класичними методами (TopHat, MaxMean та MaxMedian), що дає змогу зменшити ймовірність хибної тривоги при заданій ймовірності виявлення цілі. Тому розглянемо його принцип роботи.

3.1 Модель патч-зображення

Метод ІРІ використовує, модель зображення, яка відрізняється від традиційно прийнятої (1.1). В цій моделі, зображення перетворюється на патчі – частинки довільного розміру, на які поділене зображення. Нова модель описує ціль і фон, як патч-зображення T і B . На основі формули (1.1) створюється нова модель зображення D .

$$D = B + T + N, \quad (3.1)$$

де N – патч-зображення шуму.

3.2 Процедура отримання патч-зображень

Щоб отримати необхідне патч-зображення, вхідне зображення отримане з відеокамери ділиться на патчі, за допомогою ковзного вікна, яке має свої розміри. Розміри ковзного вікна та його крок обираються згідно поставлених завдань. Після чого з отриманих патчів отримуються вектори, для нового зображення (рис. 3.1).

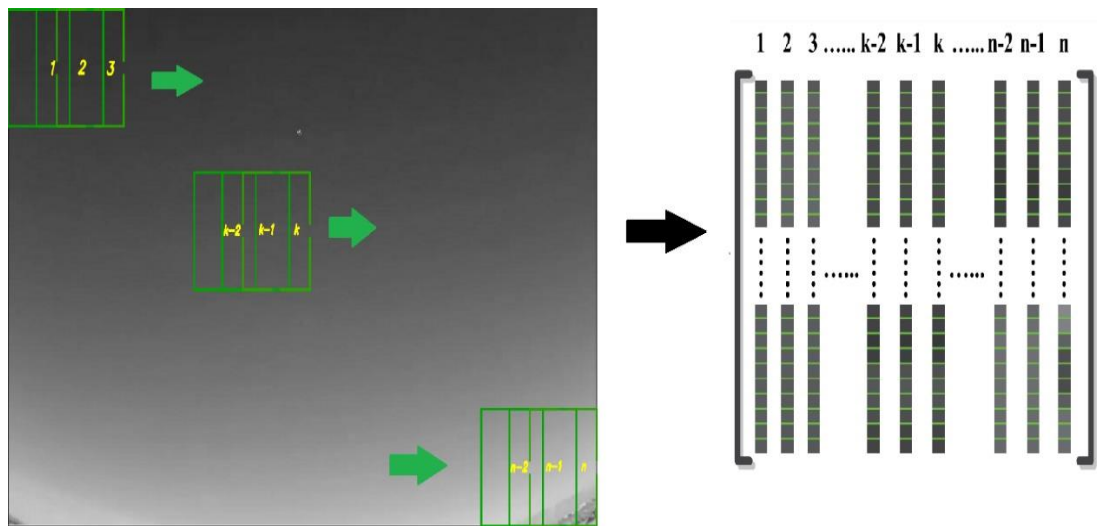


Рисунок. 3.1. Отримання патч-зображення з оригінального зображення та розбиття його на вектори

В процесі отримання патч-зображень, патчі накладаються один на одного, це означає що єдиному положенню пікселя буде відповідати декілька значень. Тому для правильного відновлення зображення необхідно застосувати 1D фільтр (рис. 3.2)

$$v = f(\mathbf{x}), \quad (3.2)$$

де $v \in \mathbf{R}$, $\mathbf{x} \in \mathbf{R}^p$ – вектор, який містить відповідні значення з p патчів. Сам 1D фільтр може бути різним і залежати від задач, які вирішуються. Наприклад: $v = \text{median}(\mathbf{x})$, $v = \text{mean}(\mathbf{x})$, $v = \text{max}(\mathbf{x})$, $v = \text{min}(\mathbf{x})$ та ін.

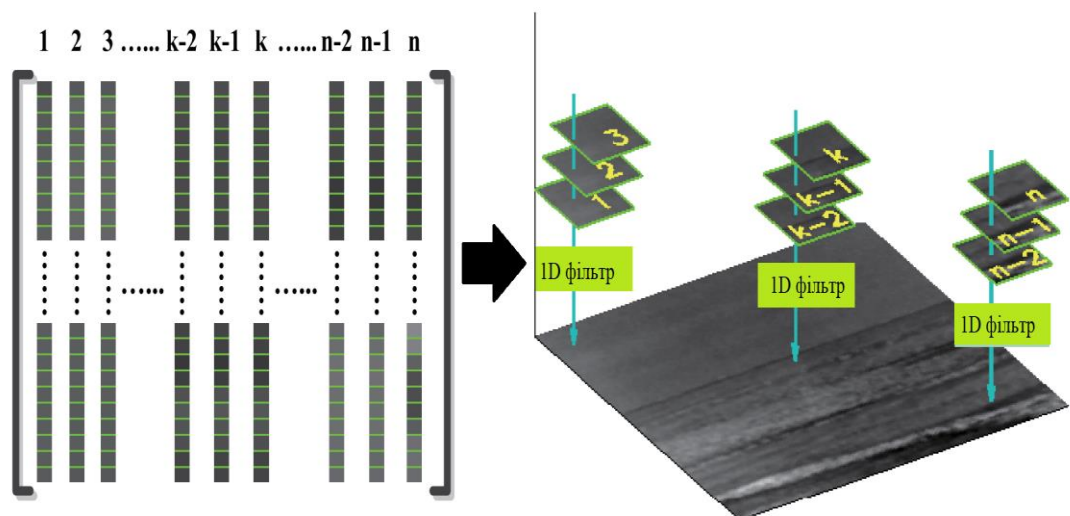


Рисунок. 3.2 Відновлення зображення

3.3 Патч-зображення цілі, фону та шуму

Оскільки виявлення відбувається точкової цілі, то така ціль є високо маневреною та може змінювати свої розміри, яскравість та контрастність в залежності від свого положення. Однак розміри цілі у порівнянні до всього зображення залишаються малими (від 2×2 до 10×10 пікселів), тому таке зображення f_T можна представити у вигляді розрідженої матриці, що означає що і патч-зображення T теж буде розрідженою матрицею

$$\|T\|_0 < k, \quad (3.3)$$

де $\|\cdot\|$ – визначає l_0 -норму, яка відповідає за лічбу ненульових елементів матриці, k – визначається кількістю цілей та їх розміром. Так як $k \ll m \times n$, де $(m \times n)$ – розміри патч-зображення цілі, то більшість елементів матриці T є нульовими.

Зображення фону часто може бути розмитим через несприятливі погодні умови, це дозволяє припустити, що сусідні і навіть несусідні патчі фону будуть мати між собою лінійну кореляцію.

У [22] був проведений експеримент з визначення сингулярних значень патч-зображень фону. Результати демонструють, що сингулярні значення з усіх патч-зображень різко зменшуються до нуля, незважаючи на те, що всі зображення були різними. Отже патч-зображення фону B представляється як низькорангова матрицю

$$\text{rank}(B) \leq r, \quad (3.4)$$

де r – константа, яка обмежує складність фону, чим більша ця константа тим складнішим є фон.

Випадковий шум вважається незалежним однаково розподіленим, також норма Фробеніуса $\|N\|_F \leq \delta$, де $\delta > 0$, тому $\|D - B - T\|_F \leq \delta$.

Отже, патч-зображення цілі є розрідженою матрицею, а патч-зображення фону - низькорангова матриця.

3.4 Алгоритм виявлення точкової цілі

Для правильного виявлення та оцінки цілі, патч-зображення необхідно отримати зображення самої цілі f_T . Це зображення отримується шляхом відновлення (реконструкції) патч-зображення цілі T . Якщо прирівняти шум до нуля і змінити формулу (3.1) на $D=B+T$, то задача виявлення точкової цілі перетвориться на задачу відтворення елементів розрідженої та низькорангової матриць. Ця задача вирішується використанням статистичного математичного апарату – методу головних компонент (Principal Component Pursuit), що вирішує таку задачу оптимізації:

$$\min_{B,T} \|B\|_* + \lambda \|T\|_1, \text{ при } D=B+T, \quad (3.5)$$

де $\|\cdot\|_*$ - ядерна норма матриці (сума сингулярних значень), $\|\cdot\|_1$ – l_1 -норма, яка обчислюється як $\|X\|_1 = \sum_{i,j} |X_{ij}|$, λ – позитивна вагова константа. $\|B\|_*$, $\|T\|_1$ заміняють $\text{rank}(B)$ та $\|T\|_0$ у формулах (3.3, 3.4) відповідно.

Такий підхід дозволяє точно і ефективно відновлювати B і T . Що дає такі переваги як: непотрібність заздалегідь оцінювати параметри різних зображень r, k з (3.3 – 3.4) та надійність виявлення точкових цілей завдяки вище згаданій оптимізації, яка може працювати не тільки з різними розмірами і рівнями яскравості цілі, але й з різними фоновими зображеннями.

Якщо наявний шум, тоді оцінювання B і T виконується за допомогою метода Stable Principal Component Pursuit .

$$\min_{B,T} \|B\|_* + \lambda \|T\|_1 \text{ при } \|D-B-T\|_F < \delta, \quad (3.6)$$

В роботі [22] ця задача була вирішена, не напямую, а за допомогою методу проксимального градієнту (Accelerated Proximal Gradient), і оптимізаційна задача дещо була видозмінена

$$\min_{B,T} \|B\|_* + \lambda \|T\|_1 + \frac{1}{2\mu} \|D-B-T\|_F^2, \quad (3.7)$$

де μ - позитивний ваговий параметр, що вказує на еквівалентність виразів (3.6) та (3.7) для деяких значень $\mu(\delta)$.

Алгоритм рішення наступний:

Вхідні дані: Матриця патч-зображення $D \in R^{m \times n}, \lambda$.

$$1: B_0 = B_{-1} = 0; T_0 = T_{-1} = 0; a_0 = a_{-1} = 1; \mu_0 > 0; \bar{\mu} > 0; \eta < 1.$$

Цикл. Поки не сходяться виконувати

$$2: Y_k^B = B_k + \frac{a_{k-1}-1}{a_k} (B_k - B_{k-1}), Y_k^T = T_k + \frac{a_{k-1}-1}{a_k} (T_k - T_{k-1}).$$

$$3: G_k^B = Y_k^B - \frac{1}{2} (Y_k^B - Y_k^T - D).$$

$$4: (U, S, V) = svd(G_k^B), B_{k+1} = US_{\frac{\mu_k}{2}}[S]V.$$

$$5: G_k^T = Y_k^T - \frac{1}{2} (Y_k^B - Y_k^T - D).$$

$$6: T_{k+1} = S_{\frac{\mu_k}{2}}[G_k^T].$$

$$7: a_{k+1} = \frac{1 + \sqrt{4a_k^2 + 1}}{2}; \mu_{k+1} = \max(\eta\mu_k, \bar{\mu}).$$

8: Перевірка умови конвергенції

$$\frac{\|D - B_{k+1} - T_{k+1}\|_F}{\|D\|_F} < \varepsilon.$$

$$9: k = k + 1.$$

Кінець циклу

Вихідні дані патч-зображення: $B = B_k, T = T_k$

3.5 Короткий опис методу ІРІ

Спочатку виконується розбиття оригінального зображення на патчі (рис. 3.5). Виділяється патч-зображення цілі T , з оригінального зображення f_0 . Після чого виконується алгоритм оцінки низькорангової матриці патч-зображення фону B і розрідженої матрицю цілі T . Для алгоритму

вибираються наступні значення: $\lambda = \frac{1}{\sqrt{\max(m,n)}}$, $\eta = 0.99$, $\mu_0 = s_2$, $\bar{\mu} = 0.05s_4$, де s_2 , s_4 друге та четверте найбільше сингулярне значення D .

Наступним кроком є відновлення зображення фону f_B та зображення цілі f_T з патч-зображень B і T , при цьому використовується 1D фільтр (3.2).

Для зменшення рівня завад може використовуватися може використовуватися пороговий підхід для одночасного виявлення яскравих та темних об'єктів: якщо виконуються умови $f_T(x, y) > t_{up}$ або $f_T(x, y) < t_{down}$, то такий піксель можна відносити до цілі, якщо ж навпаки то до фону, де

$$t_{up} = \max(v_{min}, \mu + k\sigma), \quad (3.9)$$

$$t_{down} = \min(v_{max}, \mu - k\sigma), \quad (3.10)$$

μ і σ – середнє значення і стандартне відхилення відповідно зображення цілі f_T , k , v_{min} , v_{max} – експериментально визначені значення, а призначенням останньої є видалення хибних цілей.

Для виділення об'єктів може використовуватися алгоритм сегментації, а для досягнення кінцевих результатів – алгоритми постоброблення.

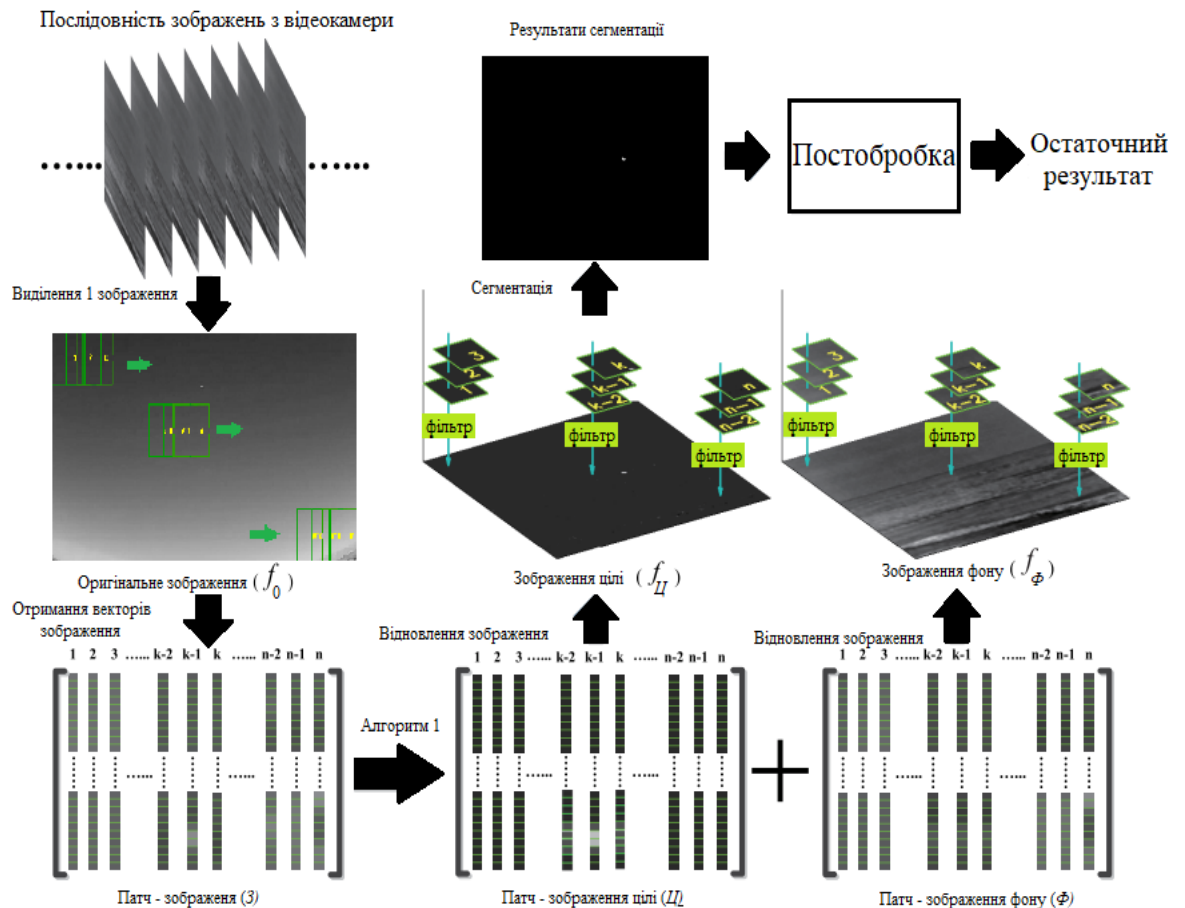


Рисунок. 3.3 Повна структурна схема алгоритму

3.6 Аналіз методу виявлення точкових цілей ІРІ

Головними параметрами алгоритму є: розмір патч-зображення, крок ковзного вікна, та константа ε . Правильний вибір вище згаданих параметрів здатен суттєво покращити якість виявлення цілі. Так як теоретично неможливо правильно визначити ці характеристики, через малі розміри цілі, її положення у просторі та ін., було проведено велику кількість експериментів, для того щоб визначити найкращі параметри.

Підібрані параметри повинні бути такі, які б задовольняли наступним вимогам, а саме:

- максимальне придушення фону та виявлення цілі,
- висока швидкодія алгоритму.

Перша серія експериментів стосується визначення *найкращого розміру патч-зображення*. Значення кроку ковзного вікна та ε задавалися

фіксованими (крок - 20×20 , $\varepsilon = 0.01$). А розміри патчу змінювалися з кроком 5.

На рис. 3.4 приведено зображення, яке підлягає дослідженню. Розмір зображення складає 1920×1080 пікселів, ціль виділена у червоному прямокутнику.



Рисунок. 3.4 Зображення, що підлягає обробленню

На рис. 3.5 приведено залежність часу виконання алгоритму від розміру патчу, а на рис. 3.6 - залежність кількості ітерацій алгоритму від розміру патчу. Також в табл. 3.1. приведено відновлені зображення цілі f_T .

Приведені результати показують, що зі збільшенням розміру патчу покращується й якість виявлення цілі. Це пов'язано з тим, що зі збільшенням розміру патчу, матриця патч-зображення цілі стає більш розрідженою. При цьому час виконання алгоритму, і кількість його ітерацій сильно зростає при збільшенні розміру патчу, що робить алгоритм повільним. Найкращий розмір патчу коливається у діапазоні від 25×25 до 45×45

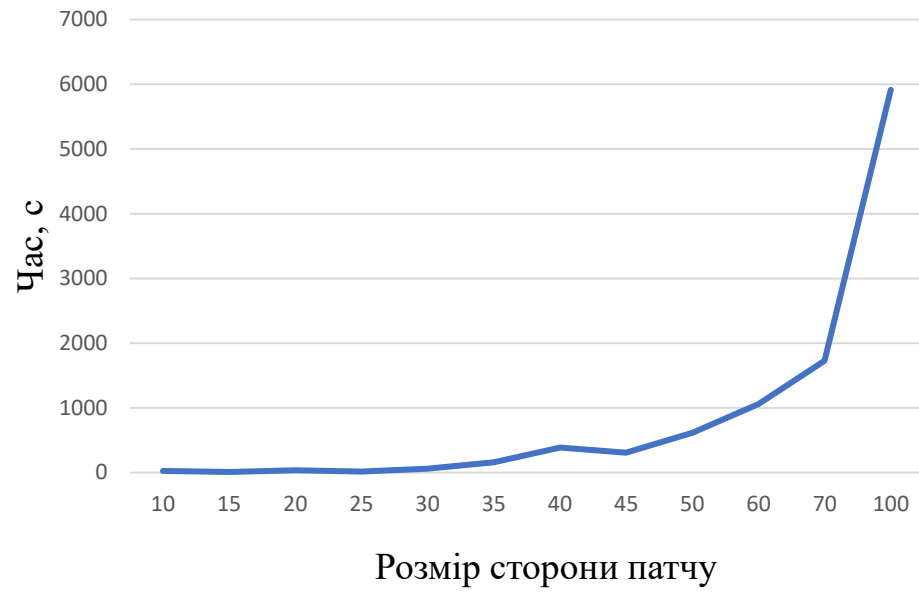
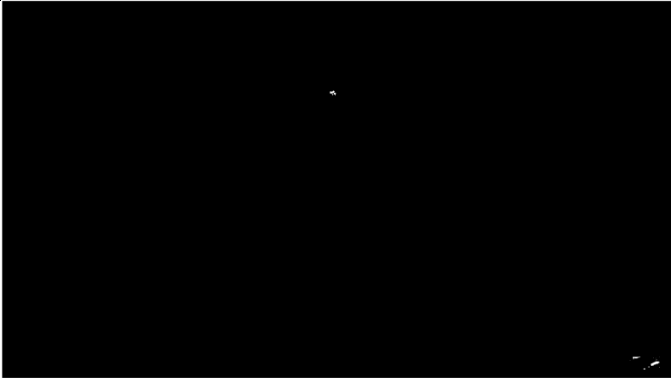
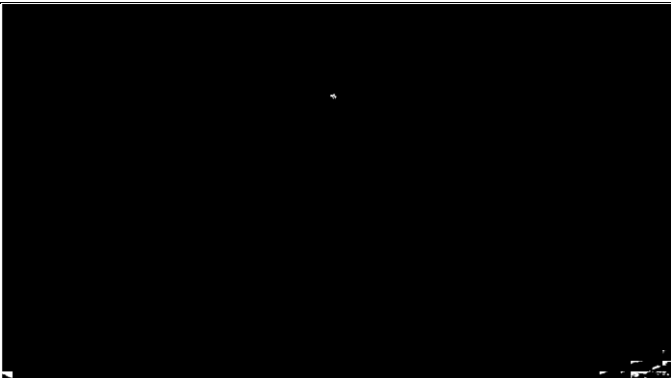
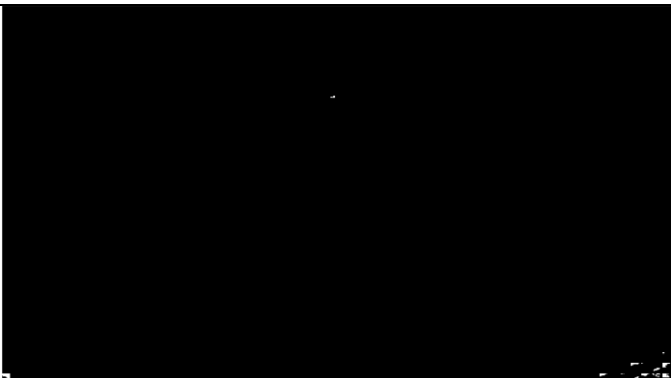

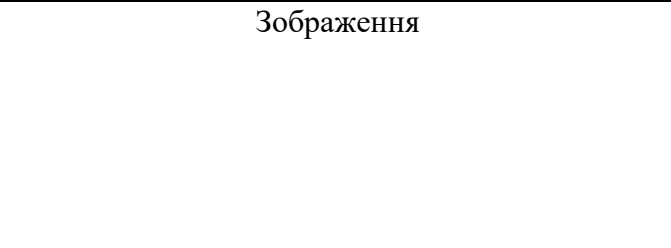


Рисунок. 3.5 Залежність часу виконання алгоритму від розміру патчу

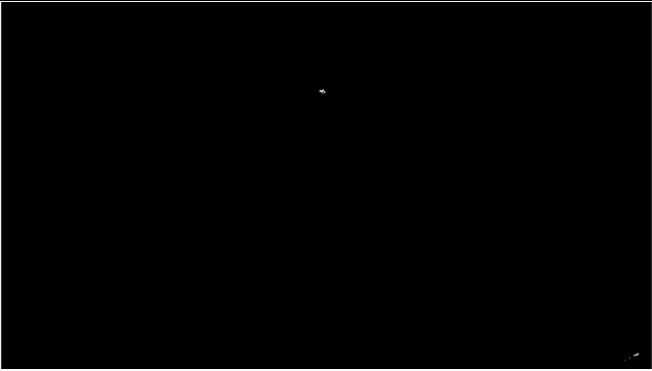
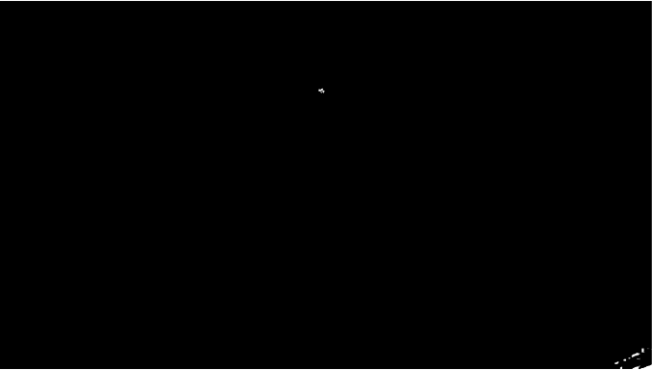
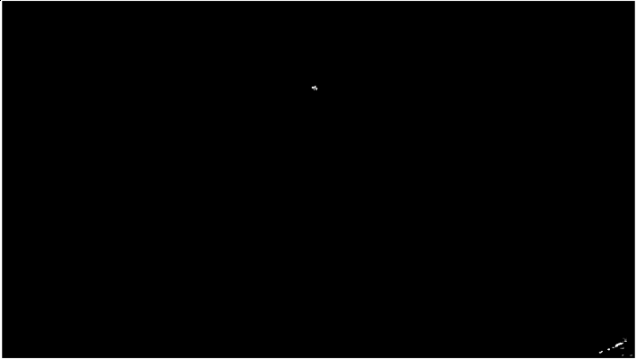
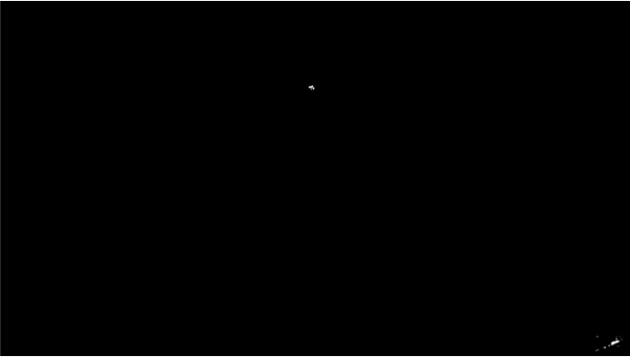


Рисунок. 3.6 Залежність кількості ітерацій алгоритму від розміру патчу

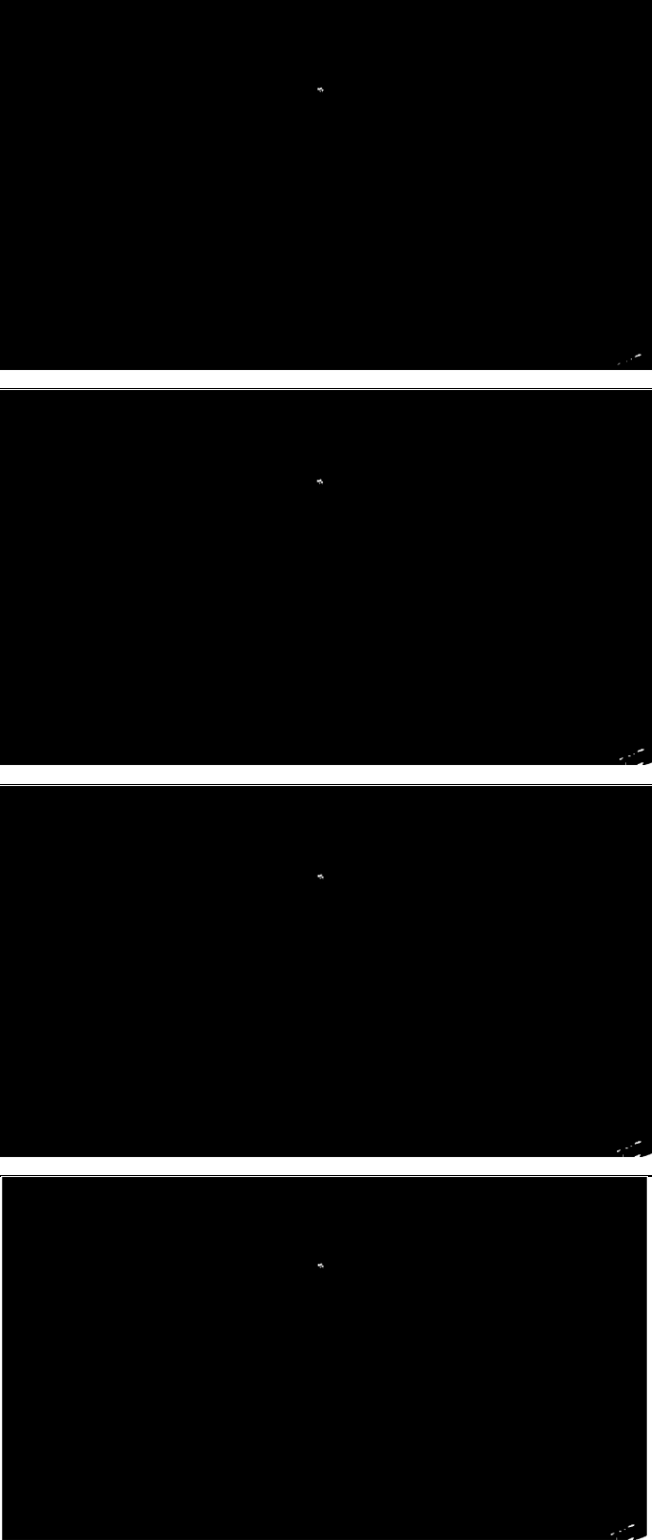
Таблиця 3.1 Результати при змінному розміру патчу

25×25	20×20	15×15	10×10	Розмір патчу
20×20	20×20	20×20	20×20	Крок ковзного
0.01	0.01	0.01	0.01	ε
13.89	34.09	9.28	26.87	Час, с
0,0099	0,0099	0,01	0,0098	Конвергенція
20	65	31	27	Кількість
				Зображення
				
				
				
				

Продовження таблиці 3.1

45×45	40×40	35×35	30×30	Розмір патчу
20×20	20×20	20×20	20×20	Крок ковзного
0.01	0.01	0.01	0.01	ε
306.22	384.73	160.98	60.96	Час, с
0.01	0.01	0.01	0,0099	Конвергенція
31	27	90	56	Кількість
Зображення				
				

Продовження таблиці 3.1

100×100	70×70	60×60	50×50	Розмір патчу
20×20	20×20	20×20	20×20	Крок ковзного
0.01	0.01	0.01	0.01	ε
5914.70	1726.50	1060.60	614.09	Час, с
0.01	0.01	0.01	0.01	Конвергенція
90	56	20	65	Кількість
Зображення				
				

Наступна серія експериментів направлена на визначення найкращого значення *кроку ковзного вікна*. Для цього бралися розміри патч-зображення 35×35 та 25×25 та змінювався крок ковзного вікна.

На рис. 3.7 приведено залежність часу виконання алгоритму від розміру ковзного вікна, при розмірі патчу 35×35 та 25×25 , а на рис. 3.8 - залежність кількості ітерацій алгоритму від розміру патчу. Також в табл. 3.2. приведено відновлені зображення цілі f_T .

З результатів цього експерименту можна побачити, що результати покращуються при використанні малого кроку, через збільшення даних, хоча й зменшується швидкодія алгоритму. Найкращий розмір ковзного вікна коливається в діапазоні 20×20 до 55×55 .

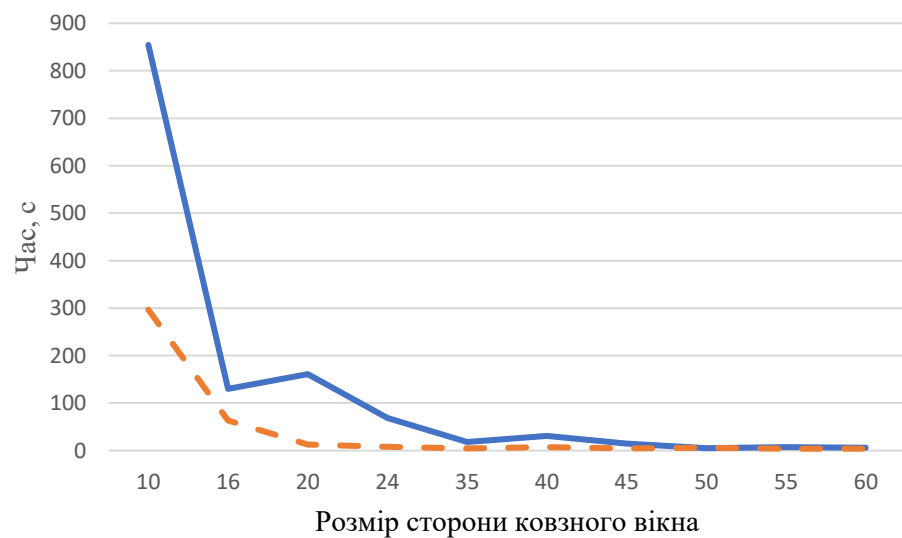


Рисунок. 3.7 Залежність часу виконання алгоритму від розміру ковзного вікна, при розмірі патчу 35×35 (суцільна) та 25×25 (штрихова)

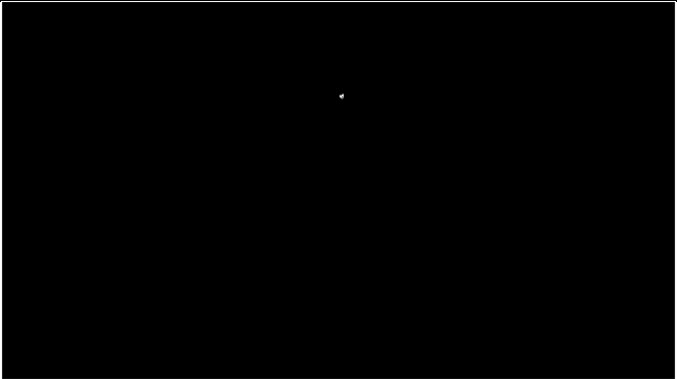


Рисунок. 3.8 Залежність кількості ітерацій часу виконання алгоритму, при розмірі патчу 35×35 (суцільна) та 25×25 (штрихова)

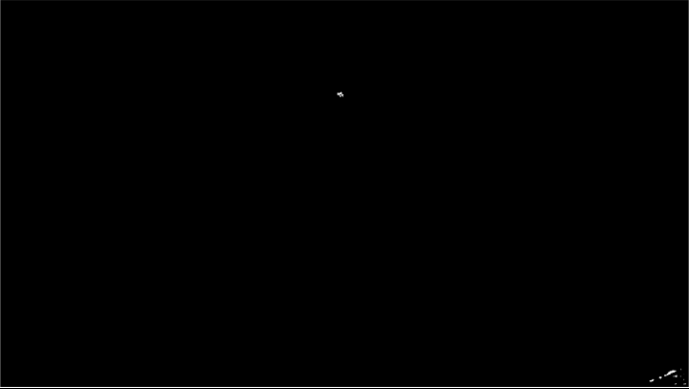
Таблиця 3.2 Результати при змінному розмірі ковзного вікна

Розмір патчу		Зображення			
Крок ковзного вікна					
ε					
Час, с					
Конвергенція					
Кількість ітерацій					
35×35	35×35	35×35	35×35	35×35	35×35
24×24	10×10	16×16	20×20	24×24	28×28
0.01	0.01	0.01	0.01	0.01	0.01
68.71	854.54	130.14	160.97	191.65	222.36
0.01	0.01	0.01	0.01	0.01	0.01
56	69	49	90	107	124

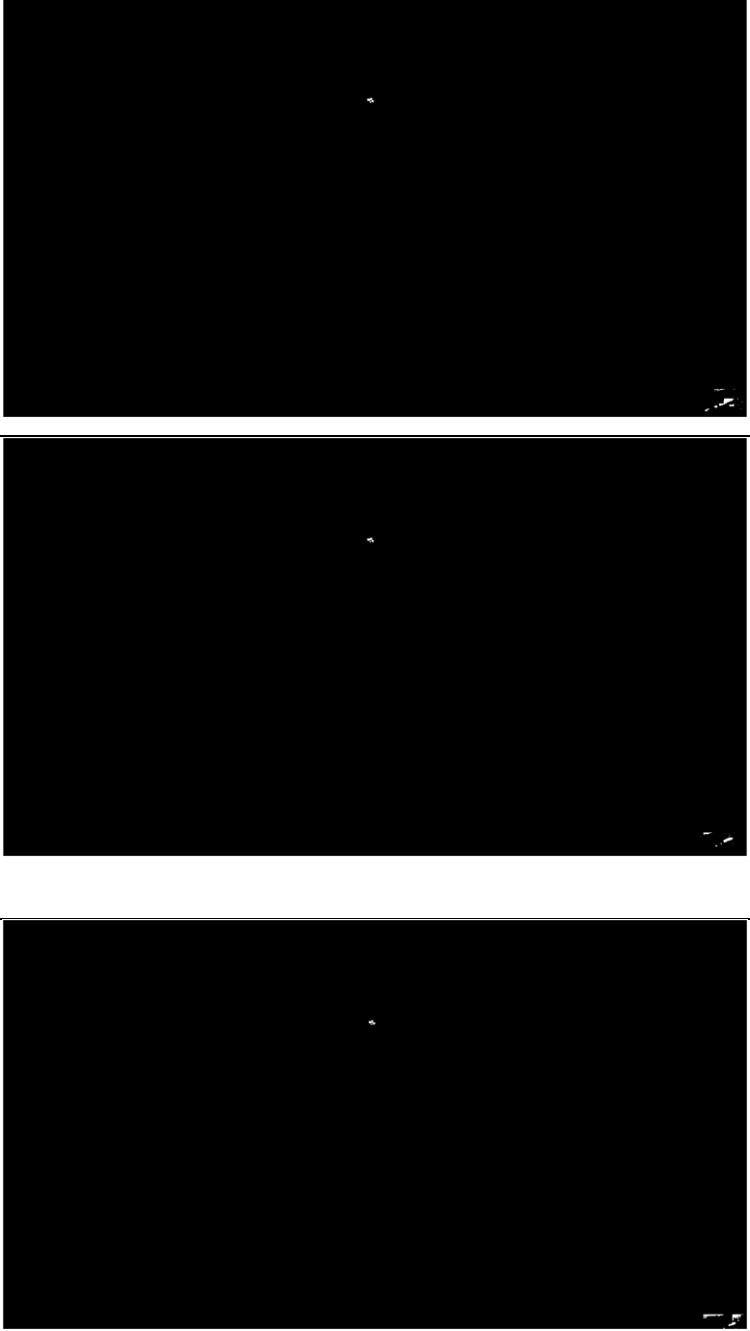
Продовження таблиці 3.2

35×35	35×35	35×35	35×35	Розмір патчу
50×50	45×45	40×40	35×35	Крок ковзного вікна
0.01	0.01	0.01	0.01	ε
5.01	14.59	30.84	18.26	Час, с
0.01	0.01	0.01	0.01	Конвергенція
23	82	122	50	Кількість ітерацій
				Зображення

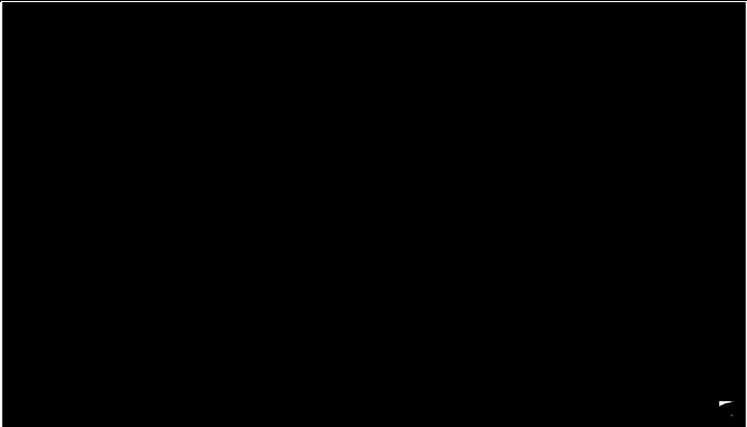
Продовження таблиці 3.2

25×25	35×35	35×35	Розмір патчу
10×10	60×60	55×55	Крок ковзного вікна
0.01	0.01	0.01	ε
296.72	6.13	7.17	Час, с
0.01	0.01	0.01	Конвергенція
45	46	51	Кількість ітерацій
			Зображення

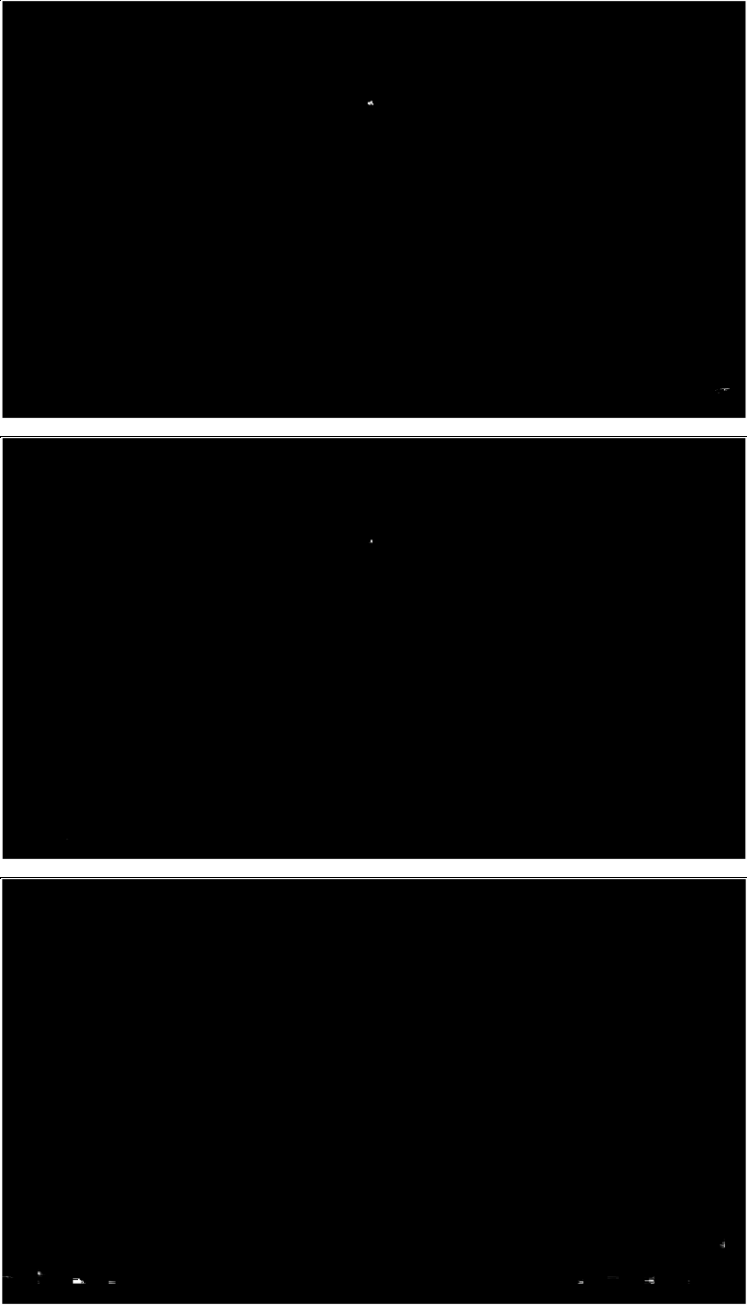
Продовження таблиці 3.2

25×25	25×25	25×25	Розмір патчу
24×24	20×20	16×16	Крок ковзного вікна
0.01	0.01	0.01	ε
7/95	12.47	63.27	Час, с
0.01	0.01	0.01	Конвергенція
20	20	50	Кількість ітерацій
			Зображення

Продовження таблиці 3.2

25×25	25×25	25×25	Розмір патчу
45×45	40×40	35×35	Крок ковзного вікна
0.01	0.01	0.01	ε
4.38	7.36	4.70	Час, с
0.01	0.01	0.01	Конвергенція
21	51	19	Кількість ітерацій
			Зображення

Продовження таблиці 3.2

25×25	25×25	25×25	Розмір патчу
60×60	55×55	50×50	Крок ковзного вікна
0.01	0.01	0.01	ε
3.87	3.88	5.72	Час, с
0.01	0.01	0.01	Конвергенція
25	23	51	Кількість ітерацій
			Зображення

Остання ж серія експериментів визначала константу ϵ при якій будуть досягнуті найкращі результати в виявленні цілі та швидкодії. Бралися різні розміри патч – зображення(25×25 , 35×35) та ковзного вікна(20×20 , 55×55). Значення ϵ змінювалися з кроком 0.05.

З результатів цього експерименту можна побачити, що зі збільшенням ϵ швидкодія алгоритму зростає, а результати виявлення падають(конвергенція збільшується).

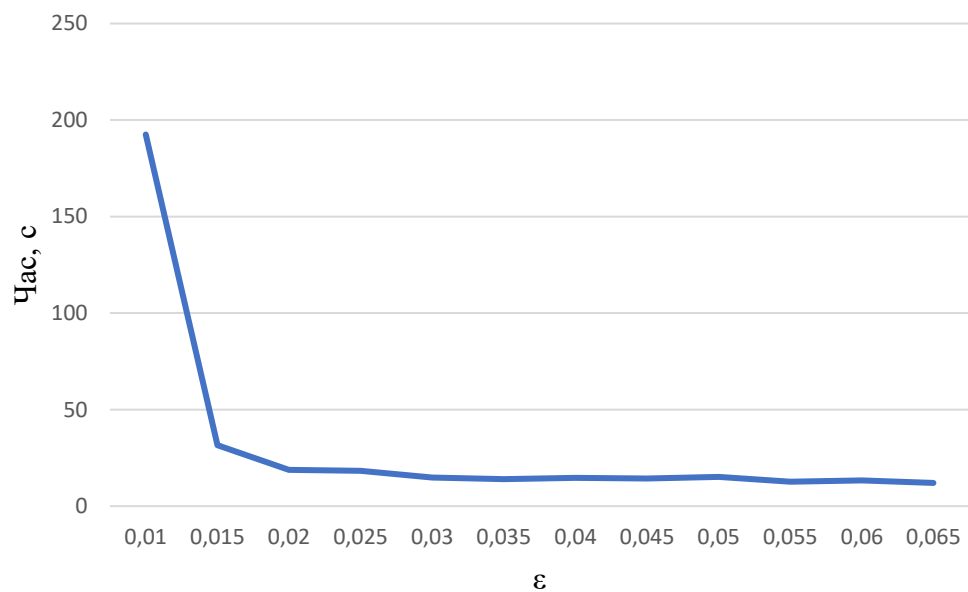
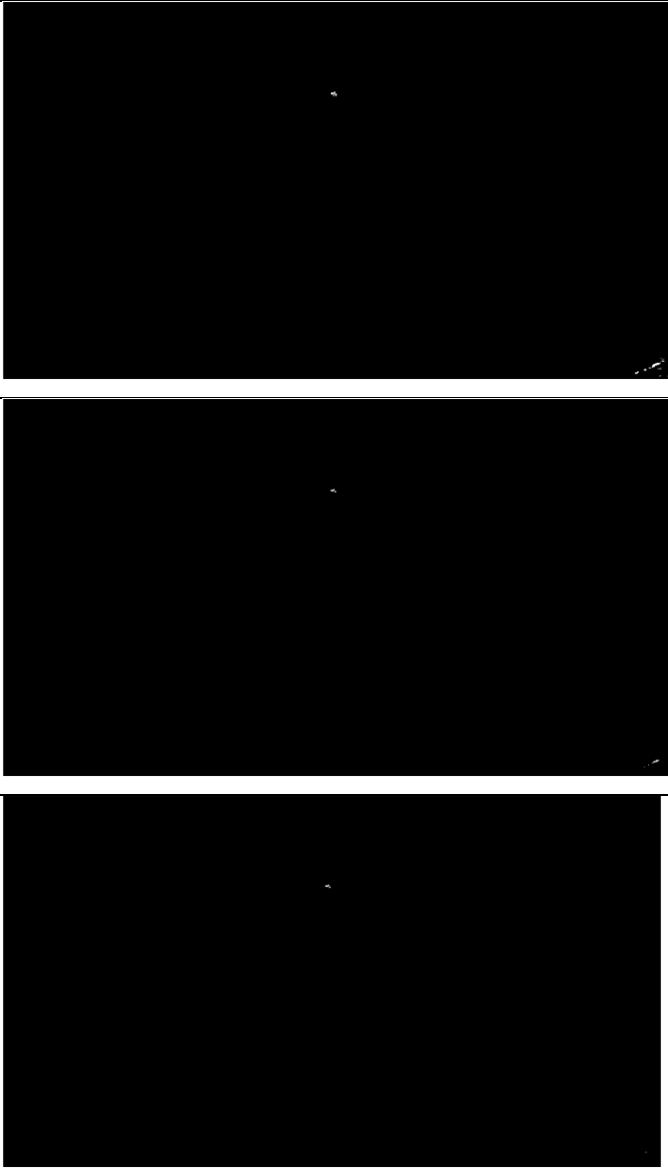
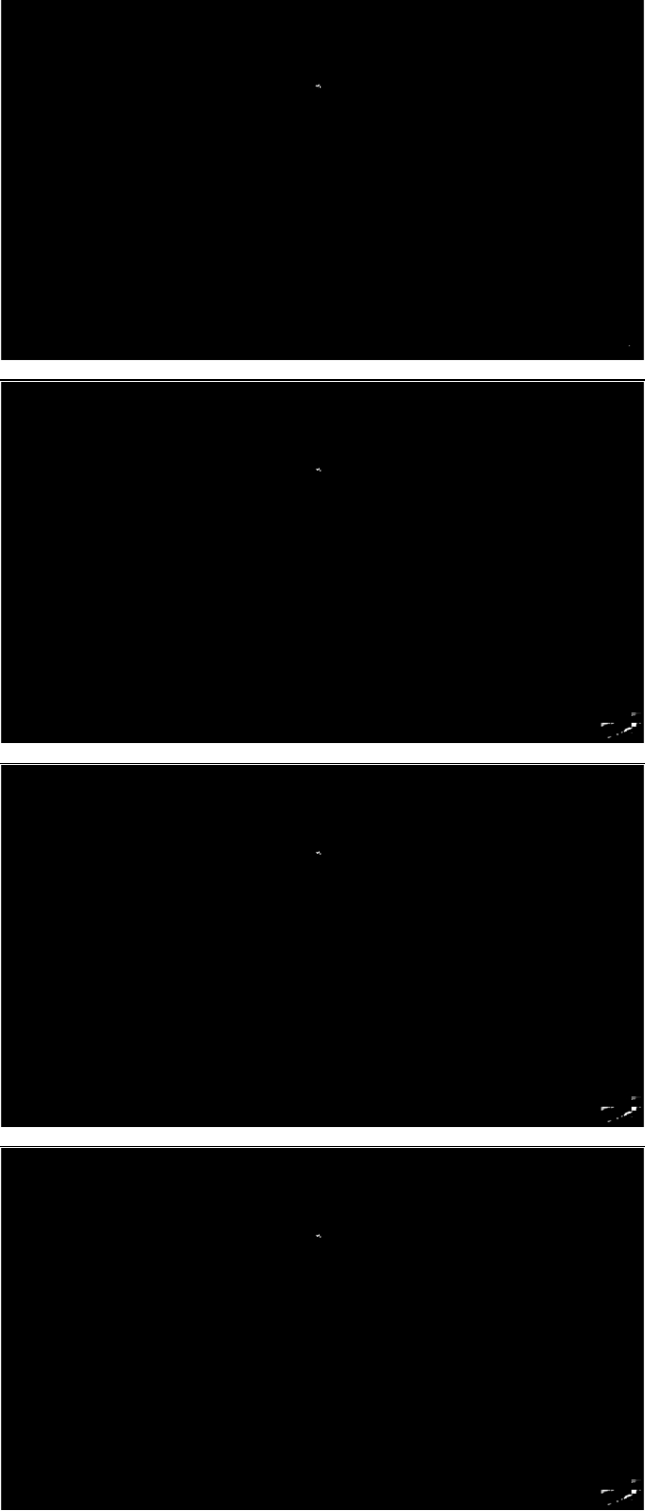


Рисунок. 3.9 Залежність часу виконання алгоритму від ϵ , при розмірах патчу 35×35 та кроком ковзного вікна 20

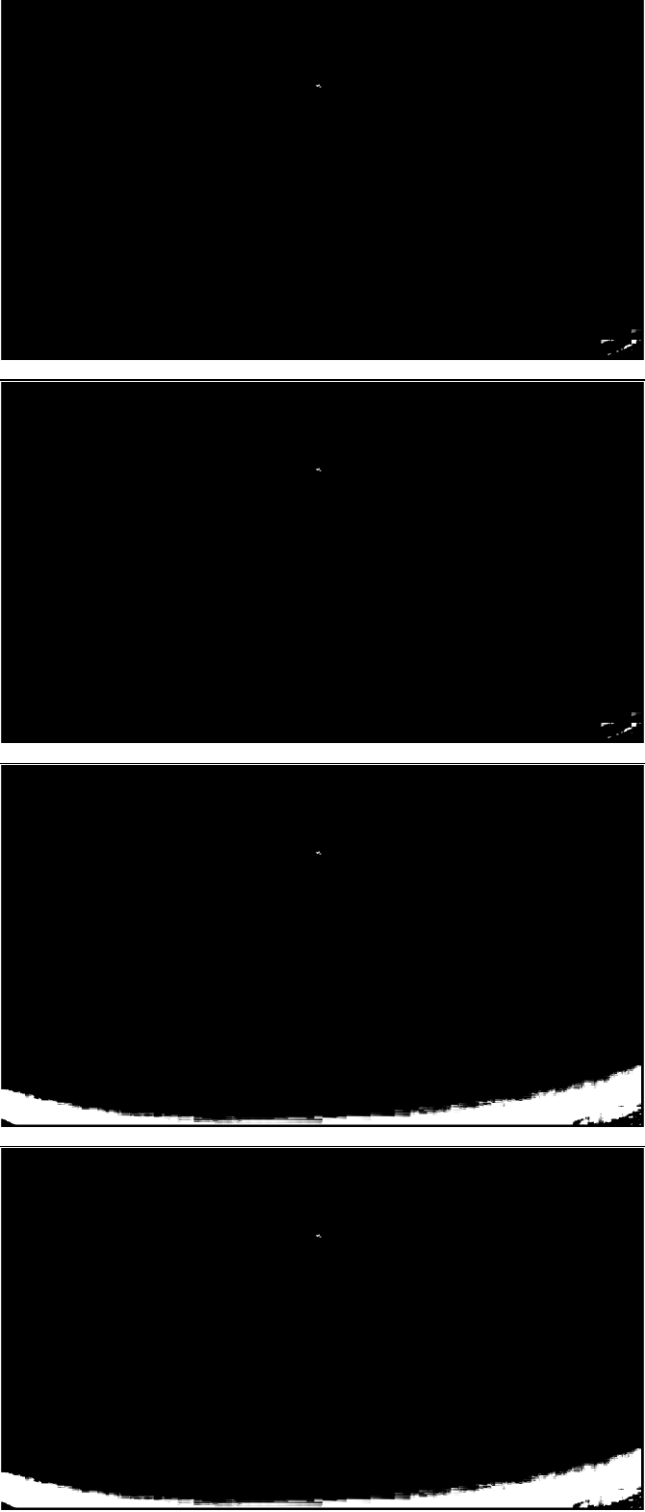
Таблиця 3.3 Результати при різних значеннях ε

35×35	35×35	35×35	Розмір патчу
20×20	20×20	20×20	Крок ковзного вікна
0.020	0.015	0.010	ε
18.80	31.47	192.52	Час, с
0.0198	0.0149	0.0100	Конвергенція
9	16	0.01	Кількість ітерацій
			Зображення

Продовження таблиці 3.3

35×35	35×35	35×35	35×35	Розмір патчу
20×20	20×20	20×20	20×20	Крок ковзного вікна
0.040	0.035	0.030	0.025	ε
14.59	13.95	14.84	18.27	Час, с
0.0272	0.0272	0.0272	0.0198	Конвергенція
7	7	7	9	Кількість ітерацій
Зображення				
				

Продовження таблиці 3.3

35×35	35×35	35×35	35×35	Зображення
20×20	20×20	20×20	20×20	
0.060	0.055	0.050	0.045	ε
13.24	12.66	15.09	14.31	Час, с
0.0540	0.0540	0.0272	0.0272	Конвергенція
6	6	7	7	Кількість ітерацій

Продовження таблиці 3.3

25×25	25×25	25×25	25×25	Зображення
55×55	55×55	55×55	55×55	
0.025	0.020	0.015	0.010	
3.39	3.47	3.70	3.92	
0.0233	0.0199	0.013	0.0095	
14	15	18	23	

Як видно з таблиць 3.1 – 3.3 найкращі результати забезпечують: розміри патчу які коливаються від 25×25 до 45×45 , крок ковзного вікна від 20×20 до 55×55 та ε від 0.01 до 0.025.

Найкращого результату вдалося досягти за таких параметрів:

- розмір патчу – 25×25 ,
- крок ковзного вікна – 55×55 ,
- $\varepsilon = 0.015$.

Час виявлення цілі склав 3.70 с.

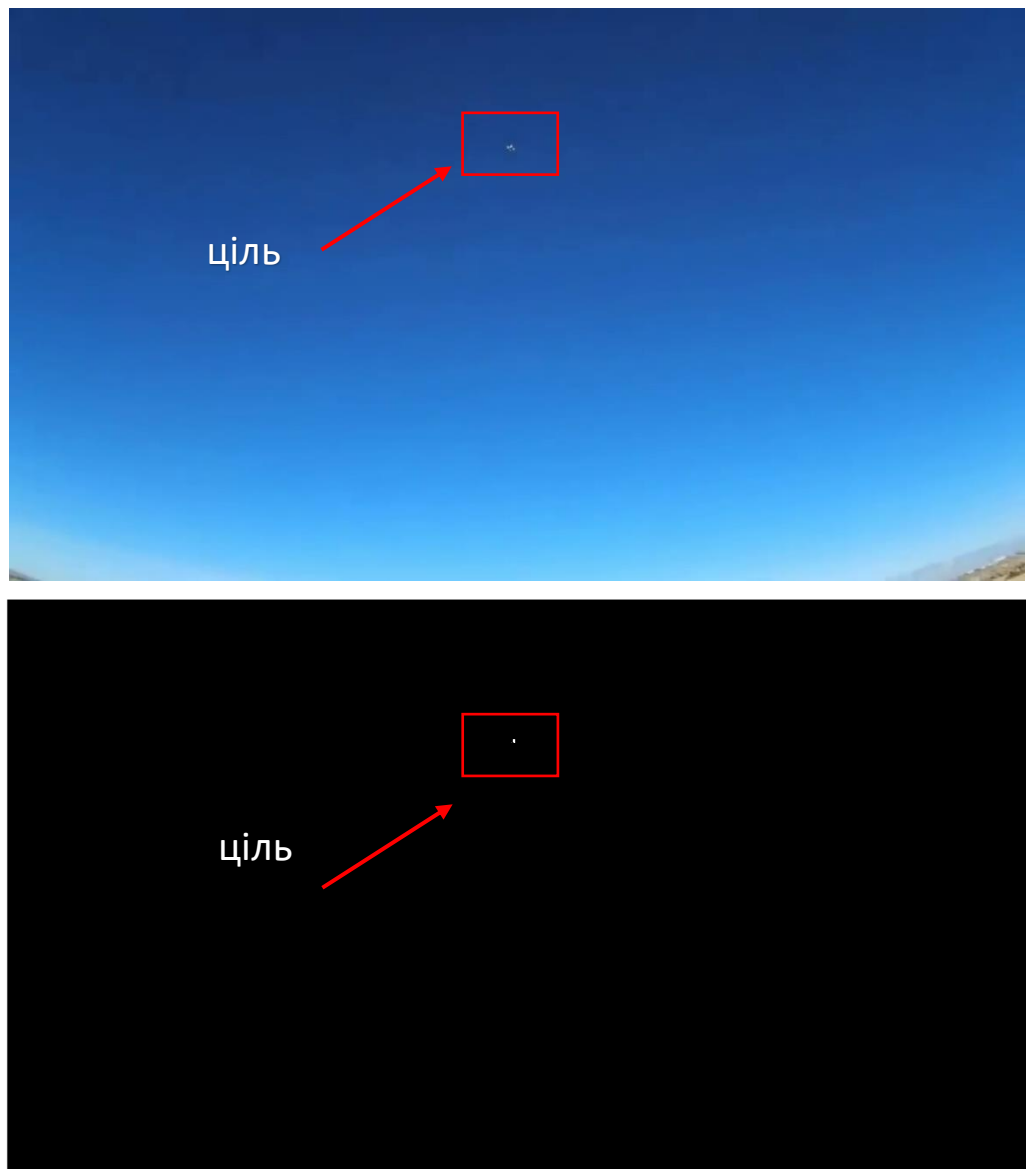


Рисунок. 3.10 Результати виявлення цілі

4. СИНТЕЗ АЛГОРИТМУ СУПРОВОДЖЕННЯ МАНЕВРУЮЧОЇ ТОЧКОВОЇ ЦІЛІ ЗА ДАНИМИ ВІДЕОСИСТЕМИ

4.1 Модель руху об'єкту

Для опису траєкторії руху об'єкту в декартовій системі координат використовують наступне рівняння [20]:

$$u_k = F_{j,k,k-1}u_{k-1} + G_{j,k}\omega_k, j = \overline{1, M}, \quad (4.1)$$

де u_k – вектор, який містить в собі параметри руху об'єкта по осям системи координат,

$F_{j,k,k-1}$, $G_{j,k}$ – матриці, які відповідають за належний маневр,

$j = \overline{1, M}$ – номер виду маневру,

ω_k – некорельований шум збудження з нульовим математичним очікуванням і кореляційною матрицею $Q_{j,k}$, $j = \overline{1, M}$.

Рівняння вимірювання системою відеоспостереження має вигляд:

$$u_k^u = Hu_k + \mathcal{G}_k, \quad (4.2)$$

де $u_k^u = (x_k^u, y_k^u)^T$ – вектор спостереження координат положення об'єкту,

$\mathcal{G}_k = (\tilde{x}_k, \tilde{y}_k)^T$ – вектор похибок вимірювання положення цілі з нульовим математичним очікуванням і кореляційною матрицею

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

Змінну переключення $a_{j,k}$, $j = \overline{1, M}$, використовують для опису типу структури моделі руху цілі (4.1), яка відповідає визначеному маневру. Ця структура (модель) відноситься до класу ланцюгів Маркова з матрицею ймовірностей переходів $\Pi_{ij,k-1,k}$, $i, j = \overline{1, M}$ і початковими ймовірностями $p_i(0)$, $i = \overline{1, M}$. Це дозволяє врахувати переходи між різними видами маневрів цілі в випадкові моменти часу.

4.2 Алгоритм супроводження цілі

Для проектування оптимального і квазіоптимального адаптивних алгоритмів траєкторної фільтрації використовують рівняння (4.1), (4.2). Результати розпізнавання виду маневру, використовують для формування оцінок вектору параметрів руху цілі.

Для вирішення поставленого завдання необхідно знайти апостеріорну щільність ймовірності процесу, що піддається фільтрації. Після чого можна визначити оцінки невідомих параметрів для вибраних критеріїв оптимальності. Так як марківський процес створюється неперервнозначним u_k і дискретними $a_{j,k}$, можна показати, що оптимальний алгоритм обчислення апостеріорної щільності ймовірності описується системою рекурентних рівнянь [20]:

$$W^*(u_k, a_{j,k}) = \sum_{i=1}^M \Pi_{ij,k,k-1} \int_{-\infty}^{\infty} \Pi(u_k | u_{k-1}, a_{j,k}) W(u_{k-1}, a_{j,k-1}) du_{k-1}, \quad (4.3)$$

$$W(u_k, a_{j,k}) = P(u_k^u | u_k) W^*(u_k, a_{j,k}) / P(u_k^u | U_{k-1}^u), \quad (4.4)$$

де $W^*(u_k, a_{j,k})$ – екстрапольована щільність ймовірності змішаного процесу,

$W(u_k, a_{j,k})$ – апостеріорна щільність ймовірності змішаного процесу,

$P(u_k^u | u_k)$ – функція правдоподібності, яка визначається рівнянням на основі (2.2).

Оптимальний алгоритм екстраполяції на один крок описується рівнянням (4.3). Екстрапольована щільність ймовірності $W^*(u_k, a_{j,k})$ уточняється рівнянням (4.4), після чого визначається апостеріорна щільність ймовірності $W(u_k, a_{j,k})$ базуючись на отриманих результатах вимірювань.

Через те, що в режимі реального часу оптимальний алгоритм фільтрації вимагає значних обчислювальних можливостей, постає потреба в розробці адаптивного квазіоптимального фільтру.

Квазіоптимальний адаптивний алгоритм фільтрації параметрів руху об'єкта, який маневрує можна описати наступними математичними співвідношеннями, якщо застосувати метод гаусівської апроксимації апостеріорної щільності ймовірності та використати методику [23].

$$W_{j,k}^* = \sum_{i=1}^M \Pi_{ij,k,k-1} W_{i,k-1}, \quad (4.5)$$

$$u_{j,k}^* = \sum_{i=1}^M \Pi_{ij,k,k-1} W_{i,k-1} F_{j,k,k-1} u_{i,k-1} / W_{j,k}^* \quad (4.6)$$

$$P_{j,k}^* = \sum_{i=1}^M [F_{j,k,k-1} \hat{P}_{j,k-1} F_{j,k,k-1}^T + G_{j,k} Q_{j,k} G_{j,k}^T + \quad (4.7)$$

$$+(F_{j,k,k-1} u_{i,k-1} - u_{j,k}^*)(F_{j,k,k-1} u_{i,k-1} - u_{j,k}^*)^T] \Pi_{ij,k,k-1} W_{i,k-1} / W_{j,k}^* ;$$

$$K_{j,k} = P_{j,k}^* H_{j,k}^T (H_{j,k} P_{j,k}^* H_{j,k}^T + R_k)^{-1}, \quad (4.8)$$

$$\hat{P}_{j,k} = P_{j,k}^* - K_{j,k} H_{j,k} P_{j,k}^*, \quad (4.9)$$

$$u_{j,k} = u_{j,k}^* + K_{j,k} (u_k^u - H u_{j,k}^*), \quad (4.10)$$

$$W_{j,k} = P(u_k^u | a_{j,k}, U_{k-1}^u) W_{j,k}^* / P(u_k^u | U_{k-1}^u), \quad (4.11)$$

$$P(u_k^u | a_{j,k}, U_{k-1}^u) = (2\pi)^{-1} (\det(D_{j,k}))^{-1/2} \exp\{0.5 \|u_k^u - H u_{j,k}^*\|_{D_{j,k}}^2\}, \quad (4.12)$$

$$D_{j,k} = H_{j,k} P_{j,k}^* H_{j,k}^T + R_k. \quad (4.13)$$

де $u_{j,k}^*$, $u_{j,k}$, $P_{j,k}^*$, $\hat{P}_{j,k}$ – математичні очікування та кореляційні матриці

умовних щільностей імовірності $W_{j,k}^*(u_k), W_{j,k}(u_k)$;

$K_{j,k}$ – коефіцієнти підсилення j -го каналу квазіоптимального пристрою(фільтру),

H – матриця спостереження,

$P(u_k^u | a_{j,k}, U_{k-1}^u)$ – умовна щільність ймовірності.

Головною відмінністю квазіоптимального алгоритму від оптимального є те, що обчислюються лише перший і другий моменти умовних апостеріорних розподілень. Він також задовольняє представлення апостеріорної щільності ймовірності $W(x_k)$ при переході на наступний крок фільтрації, як суму M гаусівських щільностей.

Квазіоптимальний фільтр є M каналним, нагадує оптимальний фільтр і має зворотній зв'язок між каналами. Рівняння (4.6 – 4.10), які фільтрують неперервну компоненту схожі з математичними співвідношеннями розширеного фільтру Калмана, цей фільтр можна використовувати коли об'єкт робить відомі маневри.

Якщо ціль супроводжується за наявності пасивних завад, на строб супроводження можуть потрапити, як істинні так і хибні відмітки. Для уникнення супроводження за хибною відміткою, обирається відмітка, яка найближче знаходиться до центру стробу [23]. Даний відбір відбувається за правилом:

$$\min_l d_j^l(k), \quad (4.14)$$

$$d_j^l(k) = \frac{(x^l(k) - x_j^*(k))^2}{p_{xj}^*(k) + \sigma_x^2} + \frac{(y^l(k) - y_j^*(k))^2}{p_{yj}^*(k) + \sigma_y^2}, \quad (4.15)$$

де $x^l(k)$, $y^l(k)$ – координати l -ї відмітки в стробі.

За істину обирається та відмітка, яка має мінімальне значення.

Якщо відмітка не потрапляє до стробу супроводження, можна застосувати пошук зі збільшеними розмірами стробу.

4.3 Аналіз точносних характеристик алгоритму оцінювання параметрів руху цілі з виявленням маневру

Аналіз синтезованого алгоритму супроводження маневруючої точкової цілі був виконаний за допомогою моделювання у середовищі Matlab (додаток Б). Траєкторія руху цілі формується на основі моделей

неманевруючої (nearly-constant-velocity model) [24] та маневруючої (nearly-constant-acceleration model) цілі на основі спрощеної моделі Зінгера [25, 26].

Вектор $u_k = (x_k, \dot{x}_k, \ddot{x}_k, y_k, \dot{y}_k, \ddot{y}_k)^T$ містить координати положення, швидкості та прискорення по відповідним осям СК. Початковий вектор $u_1 = (600, -45, 0, 300, -30, 0)^T$.

Матриці, що входять в неманевруючу модель руху, мають вигляд:

$$F_{k,k-1} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G_k = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & T^2/2 \\ 0 & T \\ 0 & 0 \end{bmatrix}.$$

Інтенсивність маневра визначалася СКВ шуму збудження ω_k , що характеризує випадкове прискорення цілі, а кореляційна матриця $Q_k = \text{diag}(\sigma_a^2, \sigma_a^2)$, $\sigma_{aj} = 0$.

Матриця переходу, що входить в маневруючу модель руху $u_k = F_{k,k-1}u_{k-1} + \omega_k$, має вигляд:

$$F_{k,k-1} = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Інтенсивність маневра визначалася СКВ шуму збудження ω_k , що характеризує випадкове прискорення цілі, з кореляційною матрицею

$$Q = 2\alpha\sigma_a^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix},$$

де $\alpha = 1/\tau_m$ - є оберненою величиною постійної часу маневру τ_m , і залежить від того, скільки секунд триває маневр. Відповідно приймалось $\tau_m = 0.05$ с, $\sigma_{a,x} = 30$ пікселів/с², $\sigma_{a,y} = 25$ пікселів/с².

Період надходження інформації складає $T = 1/25$ с, кількість періодів рівна 70. На інтервалі з 20 по 50 кадр використовувалася модель маневруючої цілі. На рис.4.1 показано траєкторія руху цілі у декартовій системі координат. На рис.4.2 як змінювалися параметри швидкості та прискорення в часі.

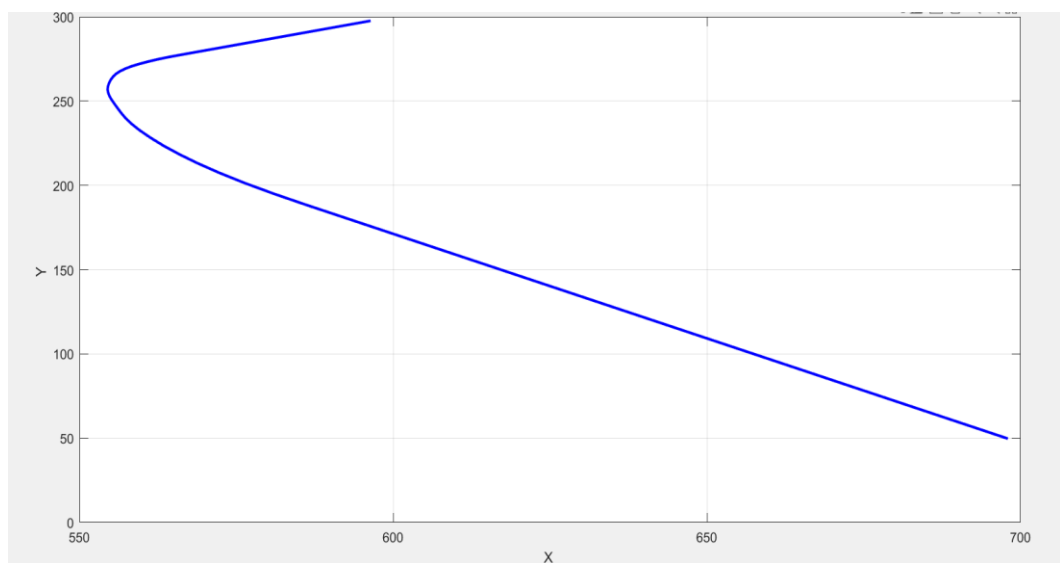


Рисунок. 4.1 Траєкторія руху цілі

Вихідні дані для алгоритму супроводження приймалися наступними. СКВ похибки вимірювання дорівнює 0.3 пікселя. З багатьох траєкторій руху об'єкта було виділено $M = 3$ найбільш характерних видів: близьке до рівномірного, повільного розвороту та маневру з максимальною перегрузкою. СКВ шуму збудження для різних типів руху рівні $\sigma_{a1} = 1$ пк/с², $\sigma_{a2} = 30$ пк/с², $\sigma_{a3} = 60$ пк/с².

Елементи матриці $\Pi_{ij,k,k-1}$ дорівнювали 0,9 при $i = j$ і 0,05 при $i \neq j$. У якості оцінок дискретного компонента вибирається гіпотеза, апостеріорна

ймовірність якої максимальна, а неперервного - відповідне умовне апостеріорне математичне очікування.

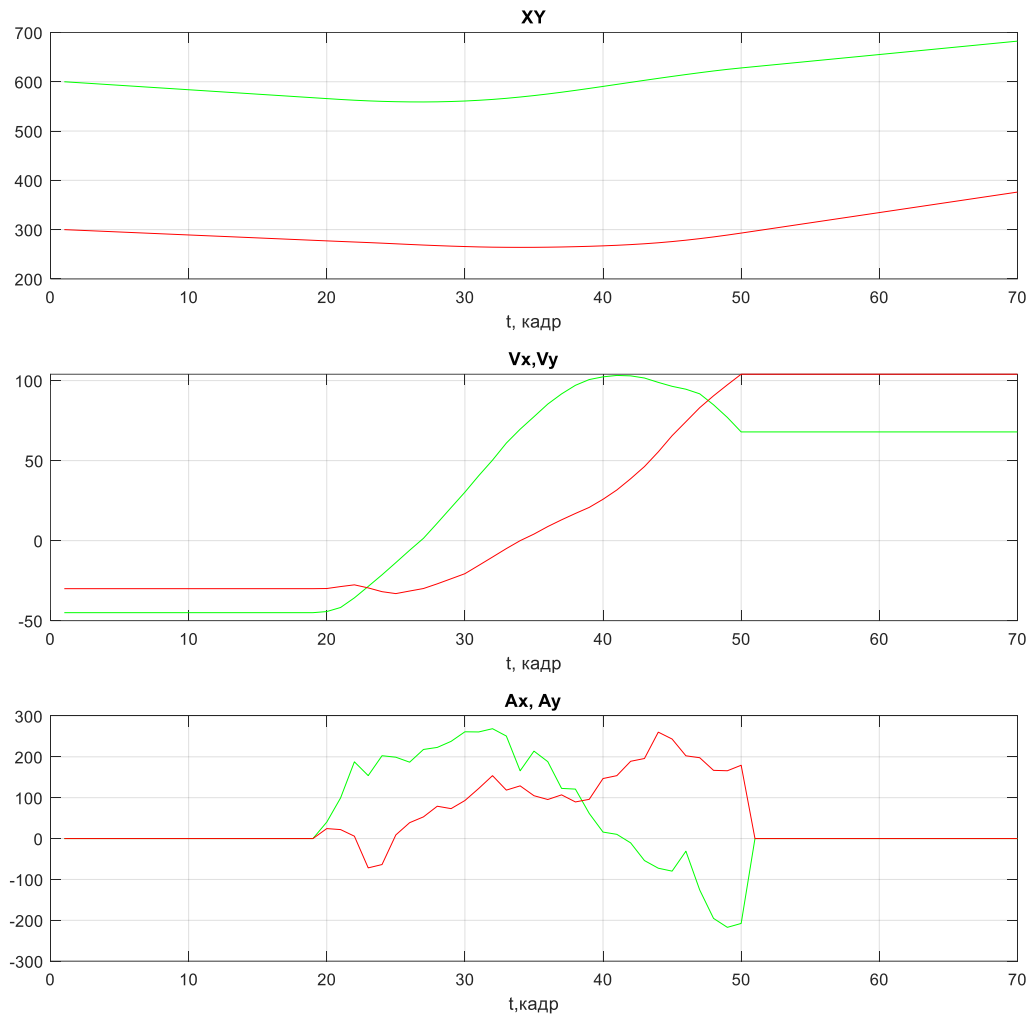


Рисунок. 4.2 Зміна положення, швидкості та прискорення цілі

Матриці, що входять в модель руху, мають вигляд:

для $M=1$:

$$F_{k,k-1} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

для $M = 2, 3$:

$$F_{k,k-1} = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & T^2/2 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix};$$

$$G_k = \begin{bmatrix} T^2 / 6 & 0 \\ T/2 & 0 \\ 1 & 0 \\ 0 & T^2 / 6 \\ 0 & T/2 \\ 0 & 1 \end{bmatrix}.$$

На рис.4.3 показаний процес виявлення типу маневру цілі. Як видно з рис.4.3 даний алгоритм точно визначає тип маневру, з поправкою на перехідні процеси, які видно на початку маневру об'єкта. Рис. 4.3 показує, що перехідні процеси в даному алгоритмі відбуваються швидко, що дозволяє точно ідентифікувати маневри.

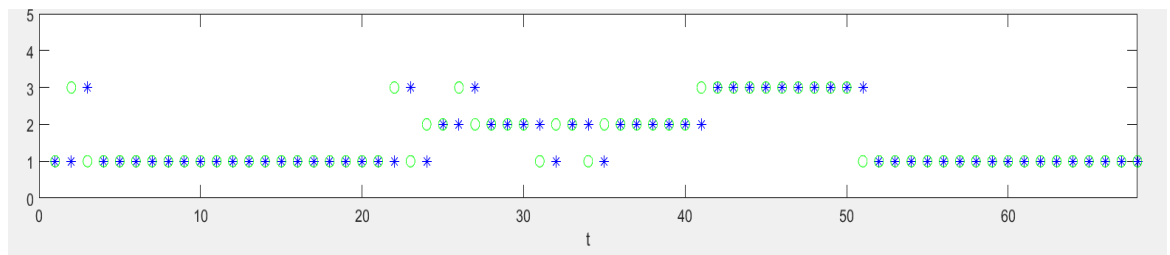


Рисунок. 4.3 Графік виявлення номеру маневру об'єкта

Наступні результати були отримані в результаті моделювання проведеного методом Монте-Карло, число випробувань – 200. Завади при цьому не моделювалися. Рис.4.4 показує залежності імовірності виявлення маневру 2-го(суцільна) і 3-го(штрихова) виду. Інтервал, протягом якого приймається рішення про зміну характеру руху, дорівнює трьом кадрам. Як видно з результатів даний алгоритм виявляє маневри 2 і 3 виду з ймовірністю що сягає 1.

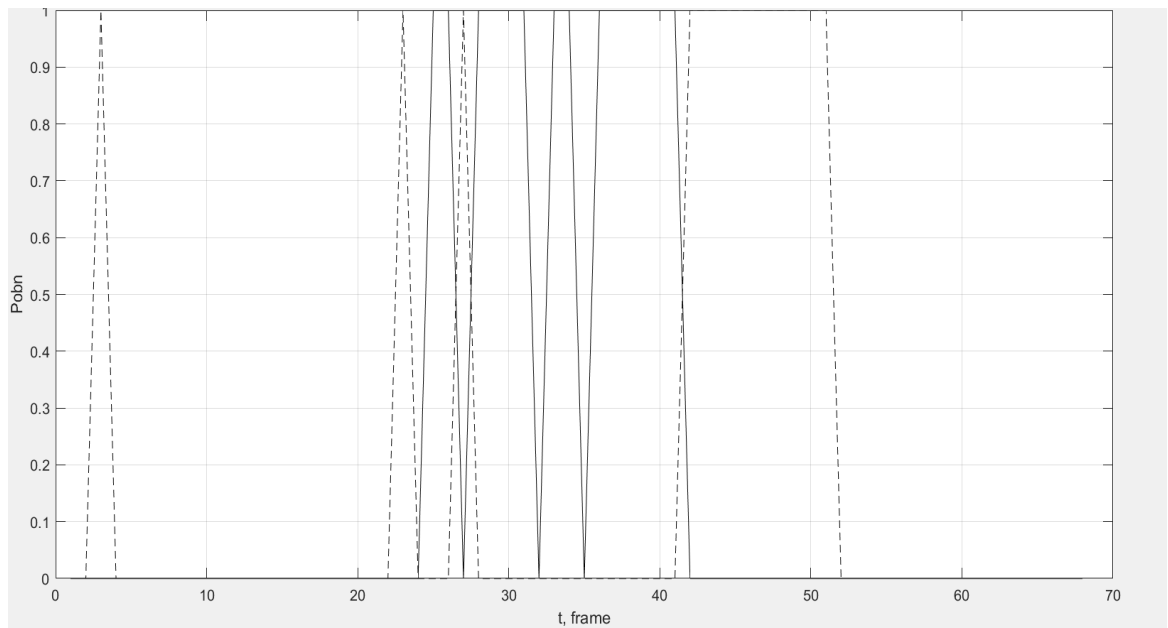


Рисунок. 4.4 Графік залежності імовірності виявлення маневру 2(суцільна) і 3(штрихова) виду

Рис.4.5 та рис.4.6 демонструють залежність СКВ, та математичне очікування помилки оцінки та прогнозу по координатам X та Y .

На рис. 4.5 штриховою лінією показане СКВ фактичних помилок прогнозу, суцільною лінією СКВ помилок прогнозу на виході фільтру, а штрих-пунктирною математичне очікування, отримані методом Монте-Карло. З графіків видно, що фактичне СКВ помилок прогнозу добре узгоджено з СКВ помилок прогнозу на виході фільтру. Математичне очікування ж коливається навколо нуля, що показує незміщеність оцінки.

На рис. 4.6 штриховою лінією показане СКВ фактичних помилок оцінки, суцільною лінією СКВ на виході фільтру, а штрих-пунктирною математичне очікування, отримані методом Монте-Карло. Графіки демонструють добре узгодження фактичного СКВ помилки оцінки з СКВ помилки оцінки на виході фільтру. Математичне очікування ж коливається навколо нуля, що показує незміщеність оцінки.

Як видно з рис.4.5 та рис.4.6 СКВ помилок оцінки та прогнозу є рівномірним і не перевищує СКВ помилок вимірювання по цих координатах.

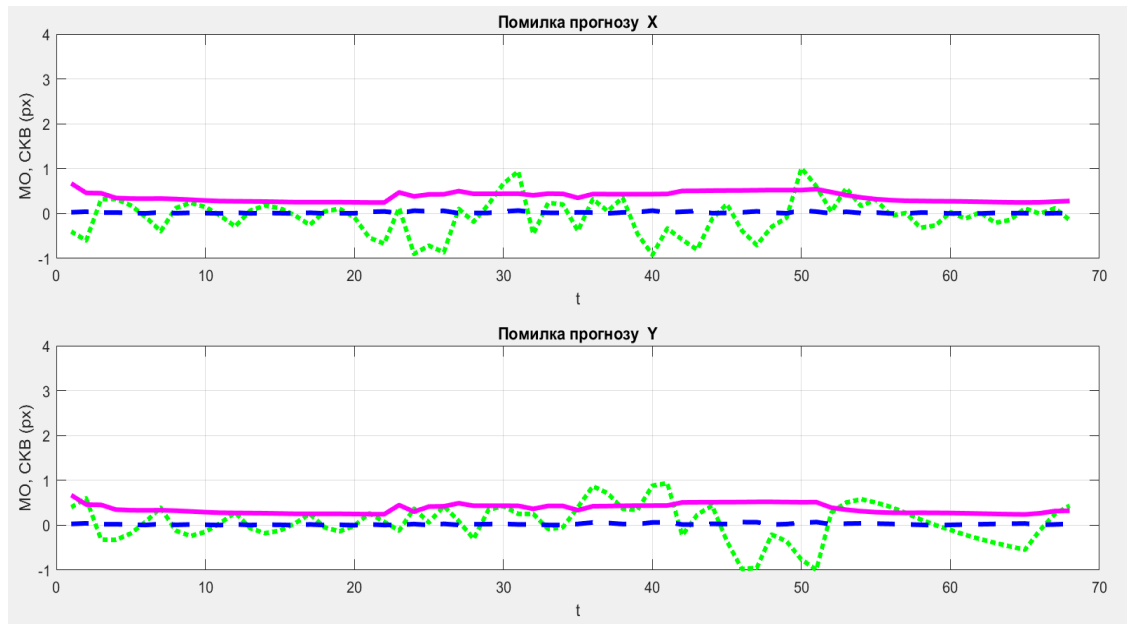


Рисунок. 4.5 Графіки залежності СКВ та мат. очікування помилки прогнозу положення цілі в декартовій системі координат

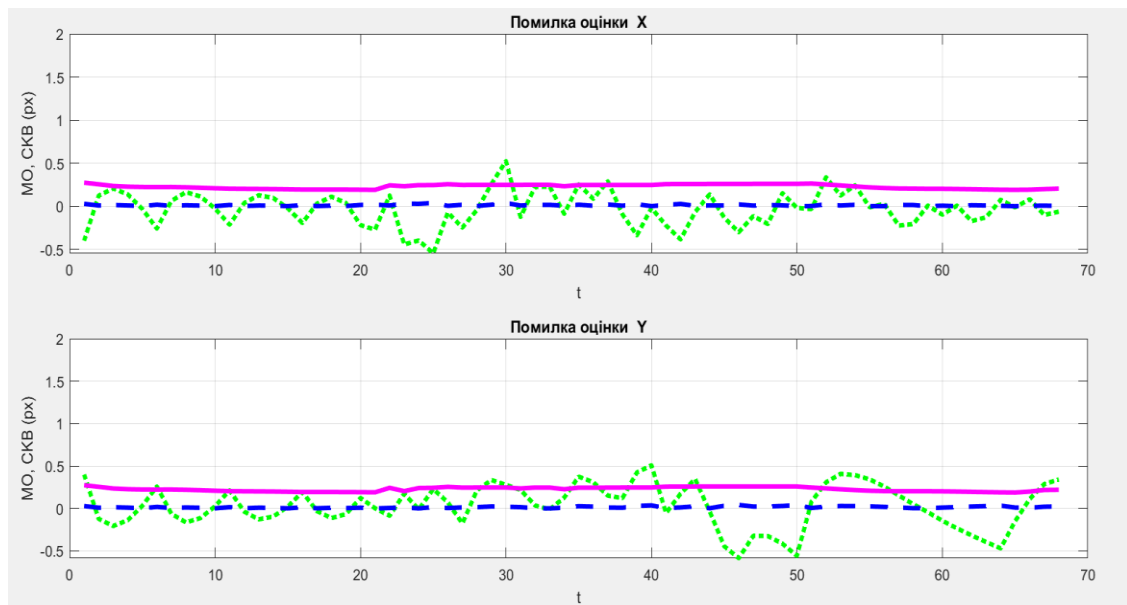


Рисунок. 4.6 Графіки залежності СКВ та мат. очікування помилки оцінки положення цілі в декартовій системі координат

Також було виконано аналіз алгоритму при наявності хибних відміток в стробі супроводження.

Припускається, що хибні відмітки виникають випадково і незалежно один від одного. Кількість цих відміток у області огляду підкоряється закону розподілу Пуассона з відомою просторовою щільністю λ

$$P_{M_k-1} = \frac{(\lambda V_k)^{M_k-1}}{(M_k-1)!} \exp(-\lambda V_k), \quad (4.16)$$

де P_{M_k-1} — ймовірність появи M_k-1 хибних відміток на k -му кроці в області розміром V_k . Координати хибних відміток в області огляду мають рівномірний закон розподілу.

На рис.4.7, 4.8 представлено приклади траєкторій зі зривом та без зриву супроводження цілі. При наявності сильних завад, до стробу супроводження можуть потрапити хибні відмітки і супроводження буде відбуватися не за ціллю, а за хибними відмітками.

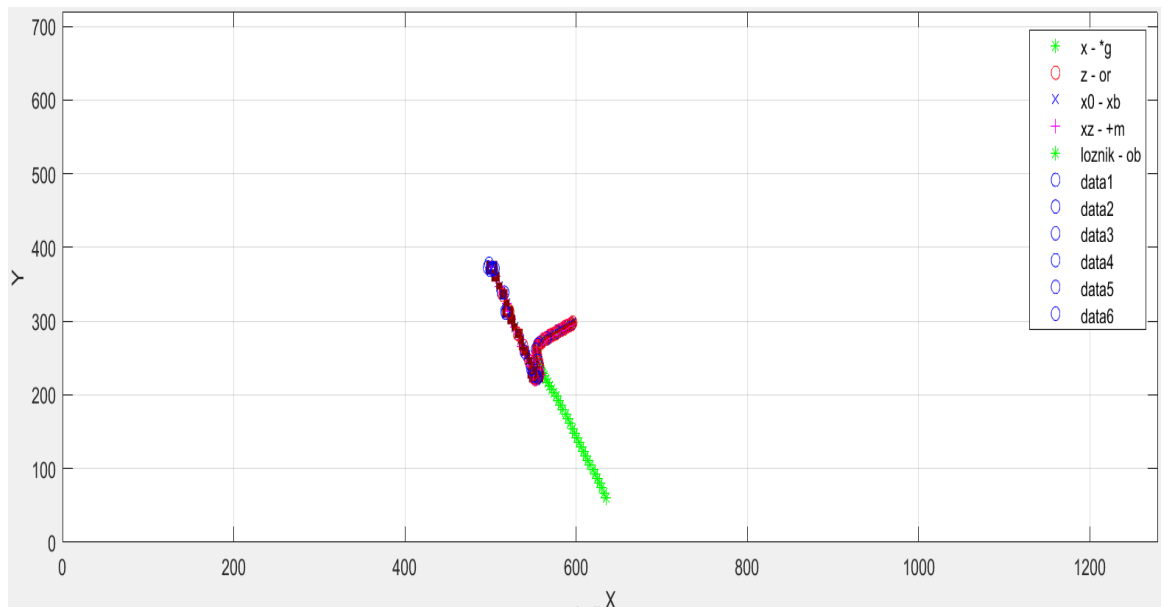


Рисунок. 4.7 Зрив супроводження за ціллю

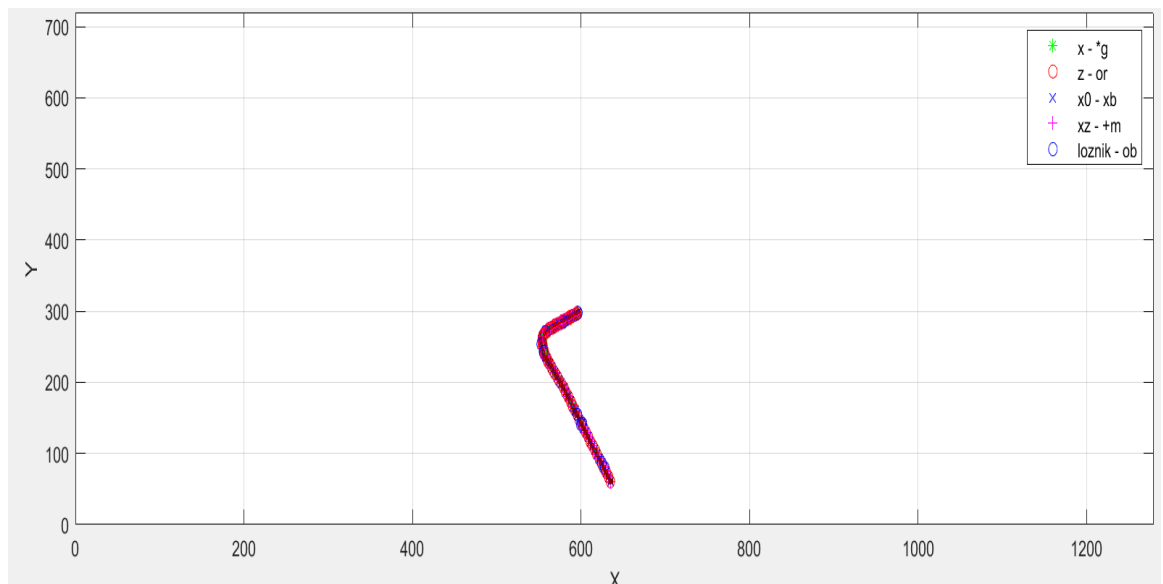


Рисунок. 4.8 Супроводження за ціллю без зриву

Як видно з рис. 4.7 зрив супроводження відбувається, при виконанні об'єктом маневру, в цей час в строб супроводження потрапляє хибна відмітка і супроводження продовжується за хибною ціллю. Зрив супроводження відбувається, якщо параметр λ стає більшим за 0.025 (середня кількість хибних відміток на зображенні розміром 1280x720 стає більшою за 23040).

5. АНАЛІЗ СИНТЕЗОВАНОГО АЛГОРИТМУ ВИЯВЛЕННЯ ТА СУПРОВОДЖЕННЯ МАНЕВРУЮЧОЇ ТОЧКОВОЇ ЦІЛІ

Для аналізу алгоритму виявлення та супроводження траєкторії маневруючої цілі було згенеровано тестову послідовність відео кадрів, за моделлю (1.1), що складається з цілі, фону і шуму. Інтенсивність та розміри точкової цілі формуються на основі (1.2), траєкторія руху цілі формується на основі моделей, що приводяться в підрозділі 4.3, і представлена на рис. 5.1

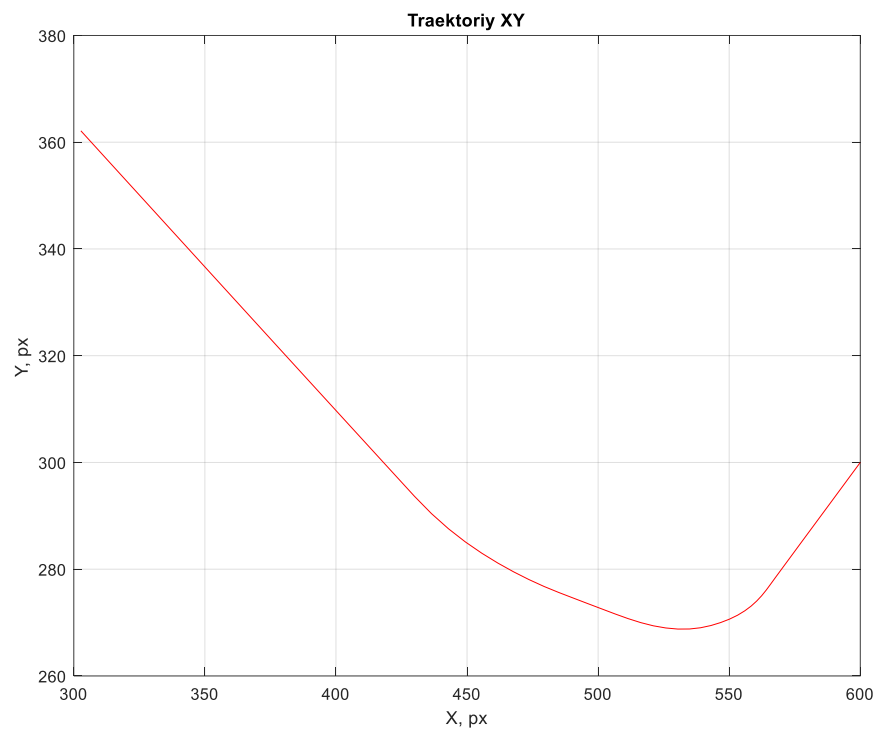


Рисунок. 5.1. Траєкторія

Приклад зображення відеопослідовності, що обробляється, показано на рис. 5.2. Розмір зображення складав 480x640 пікселів. Відношення сигнал-шум становив 20 дБ.



Рисунок. 5.2. Приклад зображення відеопослідовності, що обробляється.

Спочатку відбувається оброблення одного кадру відеопослідовності методом ІРІ (підрозділ 3.4) для виявлення точкових цілей. При цьому параметри методу приймалися наступними:

- розмір патчу – 25×25 ,
- крок ковзного вікна – 20×20 ,
- $\varepsilon = 0.2$ для збільшення швидкості оброблення кадру.

При цьому час оброблення склав в середньому 1.5 с. За допомогою оптимізації алгоритму та реалізації його на мові C/C++ є можливість цей час суттєво знизити.

На рисунках 5.3, 5.4 показані результуючі зображення після оброблення кадрів 14 та 6 методом ІРІ відповідно. Як видно з зображень ціль виділяється досить помітно, а кількість хибних відміток низька.

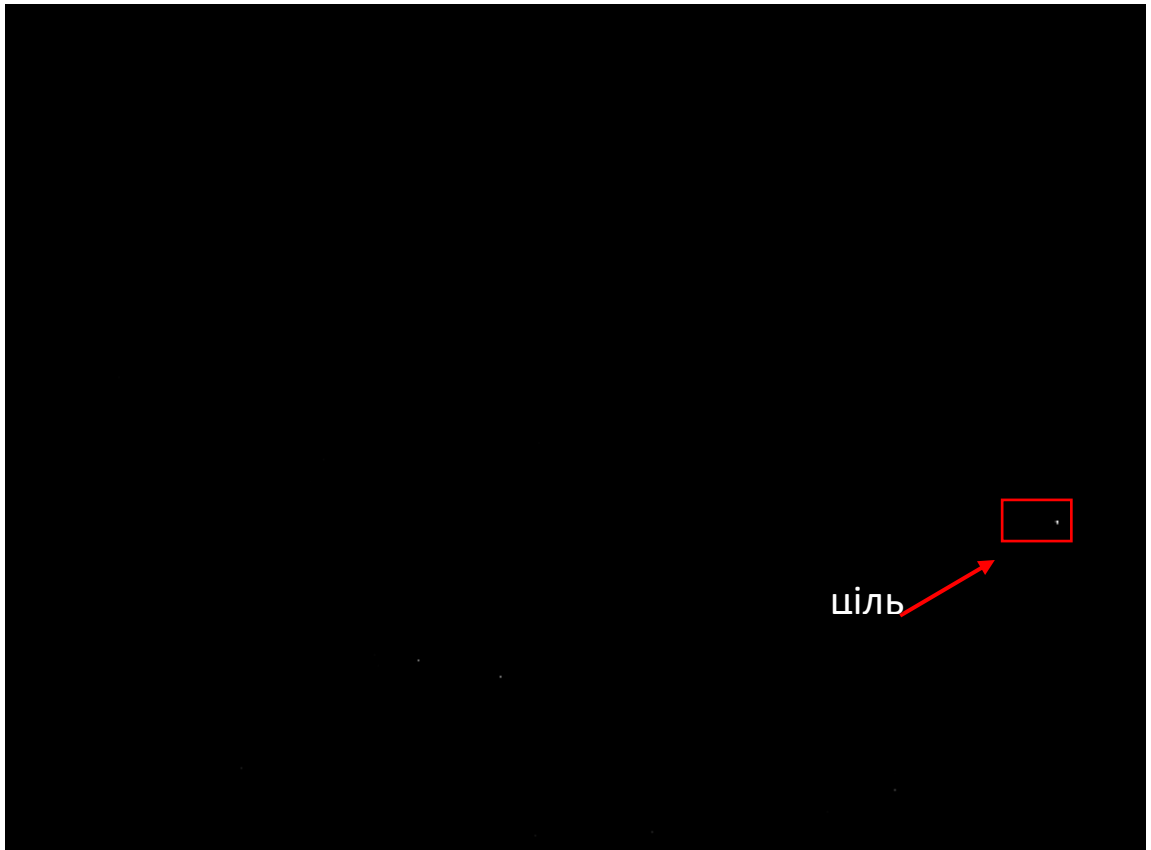


Рисунок. 5.3. Результуючі зображення після оброблення кадрів 14 методом ІРІ.

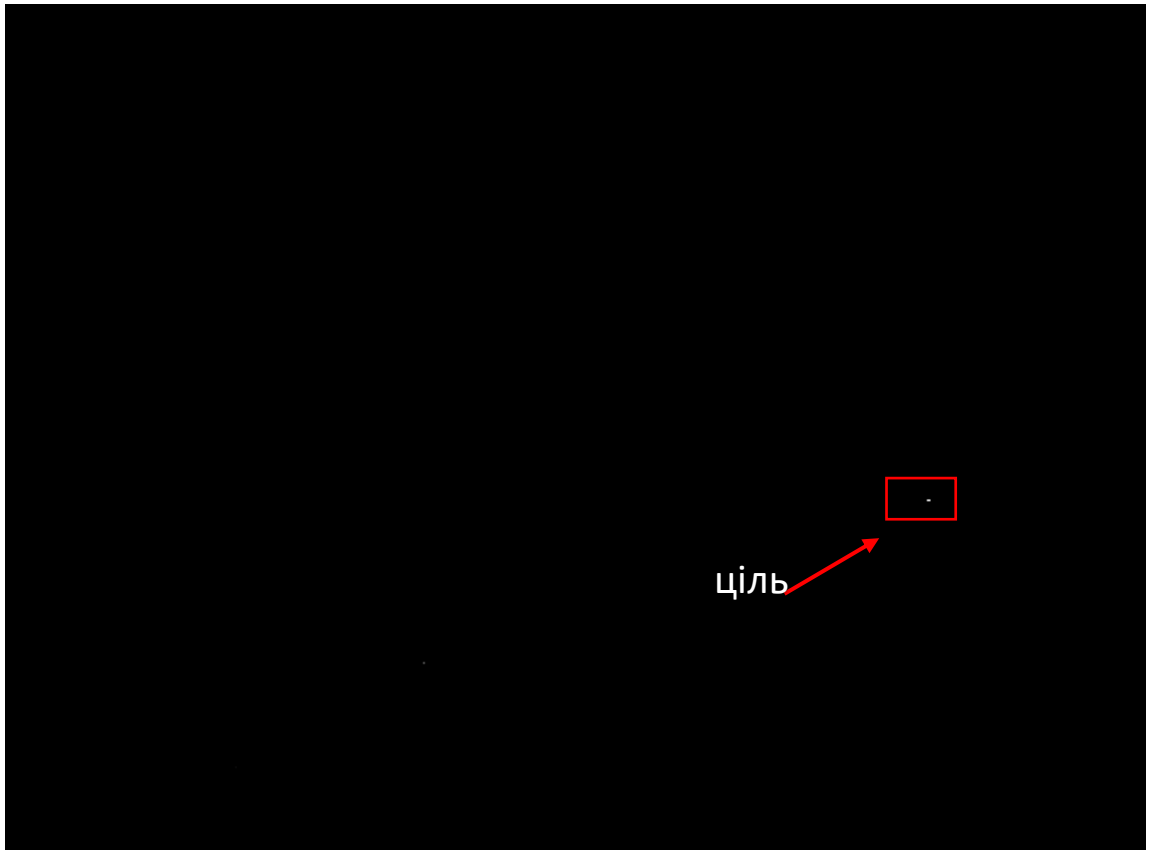


Рисунок. 5.4. Результуючі зображення після оброблення кадрів 37 методом ІРІ.

Наступним кроком є передача отриманого зображення на етап супроводження цілі, який реалізується синтезованим алгоритмом в підрозділі 4.2. Параметри алгоритму приймалися такими ж як і в підрозділі 4.3.

Ціль в стробі супроводження може займати декілька пікселів, згідно з (4.14) обирається відмітка (піксель), яка найближче знаходиться до центру стробу.

Алгоритм супроводження забезпечує оцінювання, прогнозування параметрів руху цілі, побудову стробів та траєкторії. На рис. 5.5, 5.6 представлено кадри 14 та 37 отримані даним алгоритмом. Як видно з зображень ціль знаходиться в стробах супроводження як на ділянках неманевруючого руху, так і на ділянці з маневром, що свідчить про ефективну роботу алгоритму.



Рисунок. 5.5. Кадр 14 отриманий алгоритмом супроводження



Рисунок. 5.6. Кадр 37 отриманий алгоритмом супроводження

На рис. 5.7 показано траєкторія сформована алгоритмом супроводження.

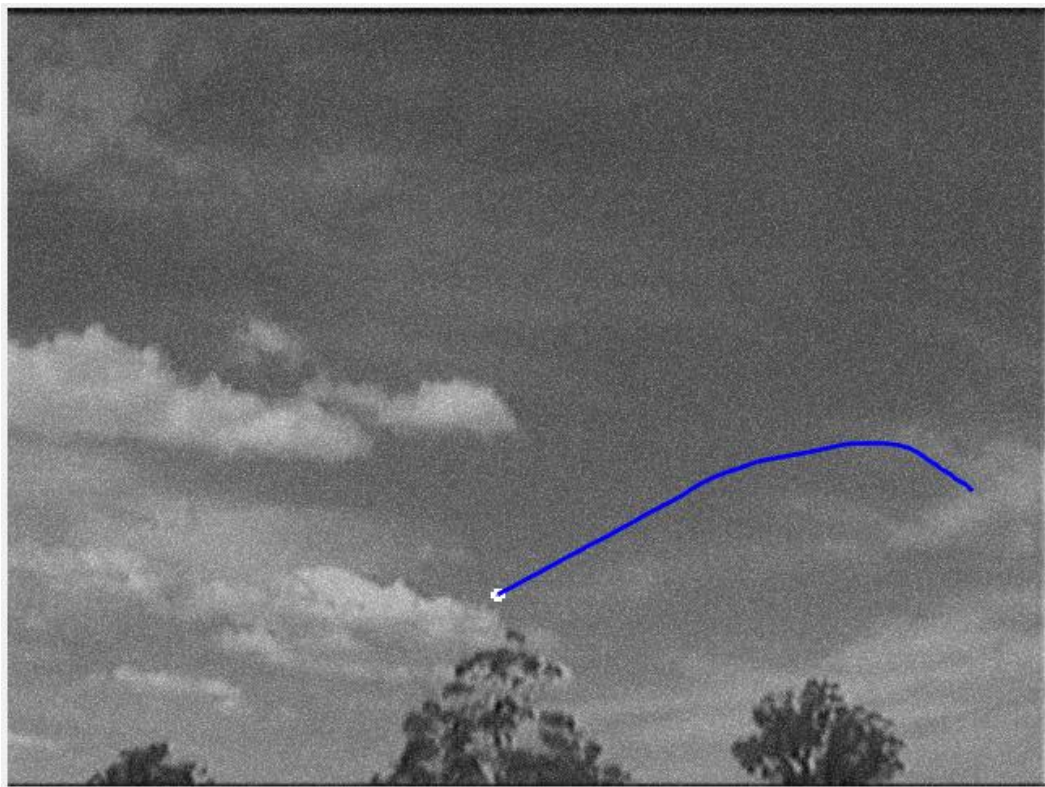


Рисунок. 5.7. Траєкторія сформована алгоритмом супроводження

Згідно з отриманих результатів використання синтезованого алгоритму виявлення та супроводження цілі на великих відстанях в сучасних системах відеоспостереження є перспективним.

Подальшими актуальними задачами є

- реалізація алгоритму на мові C/C++ та перевірка його швидкодії;
- удосконалення алгоритму супроводження з використання інформації про інтенсивність та розмірів цілі;
- проведення натурного експерименту по виявленню БПЛА системою відеоспостереження на великих відстанях.

6 РОЗРОБКА СТАРТАП ПРОЕКТУ

6.1 Ідея продукту

На основі використання морфологічного методу оберемо оптимальну ідею продукту.

За допомогою використання морфологічної карт таблиця 6.1 необхідно згенерувати ідею проекту. Система виявлення точкових об'єктів повинна швидко та безпомилково виявляти точкові об'єкти на великих відстанях в складних погодних та ландшафтних умовах за даними системи відеоспостереження.

Таким чином, до основних функції належать:

- апаратна платформа
- система відеоспостереження

Таблиця 6.1 Морфологічна карта проекту

Основні параметри	Проміжні рішення		
	1-ше	2-ше	3-ше
Апаратна платформа	Комп'ютер	Сигнальний процесор	ПЛІС
Система відеоспостереження	Фотоапарат	Відеокамера	Система відеоспостереження

На основі сформованої морфологічної карти сформуємо ідею товару: пропонується система буде виявляти та супроводжувати цілі за даними системи відеоспостереження.

Задум товару:

- Товар за задумом. Система виявлення точкових об'єктів за даними системи відеоспостереження з погляду користувача перевага перед іншими подібними системами є те, що система вже є завершеною і готовою до використання.

- Товар у реальному виконанні. Система, яка складається з системи відеоспостереження та подальшої обробки цих даних, на основі яких виконується виявлення та супроводження цілі.

— Товар з підкріпленням. Гарантійний термін 36 місяців, з технічним обслуговуванням.

6.2 План оновлення продукту

MVP(Minimum viable product) продукт з мінімальним функціоналом, який можна дати користувачам для використання. Використовується для тестування ідей у розробці програм з мінімальними затратами ресурсів, для обраного продукту зображено на рисунку 6.1

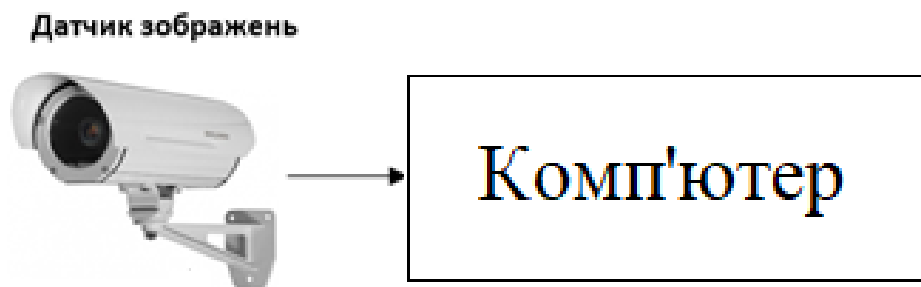


Рисунок 6.1— Мінімально життєздатний продукт

Розвиток обчислювальних засобів та відеотехніки призвів до того, що такі системи стають все більш доступними у використанні та впровадженні.

В умовах ринкової економіки планування виробничої діяльності підприємства орієнтується на максимальне задоволення попиту потенційних споживачів у продукції (роботах), послугах. План оновлення продукції наведено в таблиці 6.2

Таблиця 6.2 План оновлення продукту

№ з/п	Запитання	Відповідь
1	2	3
1	Яким цей продукт був у минулому?	Раніше застосовувались РЛС та інше
2	Частиною яких систем є продукт?	Система відеоспостереження, системи виявлення та супроводження цілі
3	Які функції надсистеми може виконувати продукт?	Продукт виконує функцію виявлення та супроводження цілі
4	Чи можна розділити продукт на частини?	Ні
5	Чи можна об'єднати (агрегувати) кілька елементів продукту в один?	Ні
6	Яким має бути ідеальний продукт?	Має мати високу точність та надійність виявлення цілі, безпомилкове супроводження за ціллю.
7	Що відбудеться, якщо вилучити цей продукт? Чим його можна замінити?	Деякі функції аналізу даних може виконувати РЛС, але це однак буде пов'язане з падінням ефективності
8	На розвиток яких функцій може бути спрямоване удосконалення продукту?	Насамперед було спрямовано увагу на удосконалення алгоритмів виявлення та супроводження за ціллю.
9	Які функції можуть залишитися «недорозвиненими»?	Функції відео системи спостереження
10	Як можна натеper розвинути ці функції?	Вдосконаленням алгоритмів виявлення, супроводження

6.3 Розроблення ринкової стратегії продукту

Продукт компанії – це пропозиція, здатна задовольнити потреби покупця. Ця ідея дуже важлива для розуміння принципу задоволеності споживача.

В таблицях 6.3 6.6 наведена сукупність маркетингових заходів, за допомогою яких компанія має намір досягнути запланованих обсягів продажу і прибутку.

Таблиця 6.3 Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Військові	Продукт є інноваційним, і цільова група більш відкрита до інновацій, ніж, в середньому, людство, тож можна припустити, що готовність буде посередньою	Орієнтований попит буде середнім	Основні конкуренти – Dedrone Drone Labs	Простота входу помірна, так як конкуренти працюють в суміжному напрямку
2	Аеропорти				
3	Організації, які займаються контролем територій, з метою запобігання проникненню чи завданню шкоди				
Які цільові групи обрано: Військові					

Таблиця 6.4 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Взяти вже існуючі відеосистеми	Концентрація на підтримці високого рівня надання послуг	Швидкість обробки інформації	Стратегія лідерства по витратах
2	Залучити до розробки алгоритмів провідні університети та науково-дослідні центри зі всього світу	Концентрація не на функціоналі системи, а на ефективності та точності алгоритмів – саме вони і є основним продуктом	Ефективність роботи алгоритмів	Диференціація

Таблиця 6.5 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Проект є першопрохідцем на ринку але є схожа реалізована ідея	Компанія буде забирати існуючих споживачів у конкурентів	Копіювати не буде, але характеристики схожі	Стратегія наслідування лідера

Таблиця 6.6 Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Точність результатів	Диференціяція	Простий у використанні	Інноваційність
2	Надійність		Доступність	Працездатність

6.4 Розроблення маркетингової програми стартап-проекту

Таблиця 6.7 Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Точність	Досягнення високої точності шляхом використання ефективних алгоритмів	Алгоритми, що планується використати, відрізняються простотою та ефективністю. Вони є результатами власних досліджень, до них не має доступу жоден з конкурентів
2	Швидкість функціонування	Використання продуктивного обладнання для досягнення високої швидкодії	Конкуренти використовують доволі продуктивне, але вже дещо застаріле обладнання, тож використання сучасного обладнання призведе до підвищення швидкості роботи власної системи у порівнянні з їхніми

Таблиця 6.8—Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Клієнти схильні до використання сучасних технологій у своїх цілях	Можливість технічної підтримки	продаж посередникам, продаж за державним замовленням	Серійним виробництвом

Таблиця 6.9—Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнти схильні до використання сучасних технологій у своїх цілях	Можливість біллінгу, тарифікація підписки	продаж посередникам в фірмові магазини	Серійним виробництвом а ПЗ у вигляді додатку

Таблиця 6.10—Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Клієнти трактують функціонал продукту та практичну цінність	Офіційний сайт	Простота використання та інтеграції; Точність результатів; Універсальність; Налаштовуваність під конкретні вимоги	Донести потенційним клієнтам інформацію про переваги нашого товару перед іншими	Ваш простір - у ваших руках

ВИСНОВКИ

В роботі була вирішена поставлена задача - синтезовано алгоритм виявлення та супроводження точкової цілі за даними системи відеоспостереження. В процесі синтезу було досліджено та проаналізовано методи та підходи з виявлення та супроводження точкових об'єктів.

Для виявлення цілі застосовувався метод ІРІ, з розбиттям оригінального зображення на патчі. Основний внесок в помилки виявлення цілі вноситься неправильним вибором розмірів патч-зображення, ковзного вікна, та константи ε . Для уникнення цих недоліків були проведені серії численних експериментів, з метою підібрати найкращі параметри.

Відповідно найкращі результати забезпечують: розміри патчу в діапазоні від 25×25 до 45×45 , крок ковзного вікна - від 20×20 до 55×55 та ε від 0.01 до 0.025. Найкращого результату для тестового зображення розміром 1280×720 вдалося досягти за параметрів: розмір патчу – 25×25 , крок ковзного вікна – 55×55 , ε – 0.015. При цьому час виявлення цілі склав 3.70 с.

Супроводження цілі здійснюється на основі синтезованого алгоритму, який являє собою набір квазіоптимальних фільтрів з зворотніми зв'язками, на основі апарату змішаних марківських процесів в дискретному часі.

Інтервал, протягом якого приймається рішення про зміну характеру руху, дорівнює трьом кадрам. Алгоритм виявляє маневри 2 і 3 виду з ймовірністю що сягає 1.

Фактичне СКВ помилки оцінки отримане методом Монте-Карло узгоджується з СКВ помилки оцінки на виході фільтра. Математичне очікування ж коливається навколо нуля, що показує незміщеність оцінки.

Зрив супроводження відбувається, якщо просторова щільність хибних відміток λ стає більшим за 0.025 (середня кількість хибних відміток на зображенні розміром 1280×720 стає більшою за 23040).

Згідно з проведеного моделювання синтезований алгоритм стабільно виявляє та супроводжує маневруючу ціль при ВСІШ 20 дБ.

В подальшому результати даної роботи можуть бути використані для створення систем: раннього виявлення об'єктів, моніторингу та контролю територій аеродромів, аеропортів та інших важливих промислових, військових та цивільних об'єктів, а також для розробки нових алгоритмів, які будуть здатні підвищити ефективність виявлення та супроводження цілей при малих ВСШ.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Arya R. A. A Comparative Study on Target Detection in Military Field Using Various Digital Image Processing Techniques / R. A. Arya, B. Radhakrishnan. // IJCSN. – 2016. – №5. – С. 181–185.
2. Yang, Huanhai. “Detection and Tracking of Infrared Dim Small Image Sequence Moving Target.” The Open Automation and Control Systems Journal 7, no. 1 (2015): 1698–1704. <https://doi.org/10.2174/18744444301507011698>.
3. Liu, X. and Zuo, Z., 2013. A Dim Small Infrared Moving Target Detection Algorithm Based on Improved Three-Dimensional Directional Filtering. Communications in Computer and Information Science, pp.102-108.
4. Маленчик Т. В. ОГЛЯД МЕТОДІВ ВИЯВЛЕННЯ ТОЧКОВИХ ЦІЛЕЙ НА ПОСЛІДОВНОСТЯХ ЗОБРАЖЕНЬ / Т. В. Маленчик, О. С. Неуймін. // IX Міжнародна науково-технічна конференція "Радіотехнічні поля, сигнали, апарати та системи" 2020. – 2020.
5. Hadzagic M. Track-Before-Detect Methods in Tracking Low-Observable Targets: A Survey / Melita Hadzagic, Hannah Michalska, Eric Lefebvre // Sensors and Transducers Magazine (ST e-Digest), Special Issue. — 2005. — P. 374-380.
6. Rawat, S. S., Verma, S. K., & Kumar, Y. (2020). Review on recent development in infrared small target detection algorithms. Procedia Computer Science, 167, 2496–2505.
7. Giuseppe B., Angelo C., Antonio P. Small Target Detection Using Wavelets. Proceedings of the International Conference on Pattern Recognition, 1998:1776-1
8. Strickland, R., Han, et al. Wavelet Transform for Detecting Microcalcifications in Mammograms. IEEE Trans. Medical Imaging, 1996, 15 (2):218-229.
9. Mallat, S. G. Theory for Multiresolution Signal Decomposition: the Wavelet Representation. IEEE Trans. PAMI, 1989, 11 (7):674-693.

10. Gregoris, D. J., et al. Detection of Dim Targets in FLIR Imagery Using Multiscale Transforms. *Proceedings of SPIE*, 1994, 2262:62-71.
11. Chen X.Z., Sun H.Y. Image Targets Detection based on the Energy Feature. *Infrared and Laser Engineer*, 2001, 30 (1):30-32.
12. Cheng Z.X. Lin Y.P. Application of wavelet analysis in image processing. *Journal of Engineering Mathematics*, 2001, 18 (5):57-86. 778
13. Bai, Xiangzhi, Shan Zhang, Binbin Du, Zhaoying Liu, Ting Jin, Bindang Xue, and Fugen Zhou. "Survey on Dim Small Target Detection in Clutter Background: Wavelet, Inter-Frame and Filter Based Algorithms." *Procedia Engineering*. December 06, 2011.
14. Wang C.B. Phd Thesis, Shanghai Jiaotong University, 2000.
15. Neri, A., Colonnese, et al. Automatic Moving Object and Background Separation. *Signal Processing*, 1998, 66 (2):219-232
16. Deshpande, S.D., Er, et al. Max-Mean and Max-Median Filters for Detection of Small-targets. *Proceedings of SPIE*, 1999, 3809:74-83.
17. Soni, T., Zeidler, et al. Performance Evaluation of 2—D Adaptive Prediction Filters for Detection of Small Objects in ImageData. *IEEE Transactions on Image Processing*, 1993, 2 (3):327-340.
18. Jiazhen Liu, Askar Hamdulla. "Overview of Infrared Dim and Small Target Detection Methods." *Web of Proceedings - Francis Academic Press*. August 28, 2019.
19. Tunç, S. and H. Ilgin. "Dim Target Detection in Infrared Images Using Saliency Algorithms." *Radioengineering* 27 (2019): 635-642.
20. Жук С.Я. Методы оптимизации дискретных динамических систем со случайной структурой / С.Я. Жук: [Монография]. — К.: НТУУ «КПИ», 2008. — 232 с.
21. Рязанцев Л. Б. Многомодельное байесовское оценивание вектора состояния маневренной воздушной цели в дискретном времени / Л. Б. Рязанцев. // *Вестник ТГТУ*. — 2009.

22. Gao, C., Meng, D., Yang, Y., Wang, Y., Zhou, X., & Hauptmann, A. G. (2013). Infrared Patch-Image Model for Small Target Detection in a Single Image. *IEEE Transactions on Image Processing*, 22(12), 4996–5009.
23. Жук С.Я. Адаптивная фильтрация параметров движения маневрирующего объекта в прямоугольной системе координат / С. Я. Жук, В. И. Кожешкурт, В. В. Юзefович // Реєстрація, зберігання і обробка даних.— 2009.— Т.11.— №2.— С.12— 24.
24. X. Rong Li, Vesselin P. Jilkov. Survey of Maneuvering Target Tracking. Part I: Dynamic Models. *IEEE Transaction on Aerospace and Electronic Systems*. Vol. 39. No 4. October 2003. P. 1333-1363.
25. Singer R. Estimating optimal tracking filter performance for manned maneuvering targets // *IEEE Trans. Aerospace & Electronics Systems*. – July 1970. – Vol. AES -6. – p. 473 – 483.
26. Зингер Р.А., Бенке К.В. Оценка характеристик и выбор фильтров сопровождения в реальном масштабе времени для тактических систем вооружения // *Зарубежная радиоэлектроника*. – 1972. – No 1. – С. 44 – 60

ДОДАТКИ

ДОДАТОК А

Програма алгоритму виявлення цілі :

```

clear vars
% Load image (matrix)
%Image = imread('Test_drone.jpg');
Image = imread('vlcsnap2.png');
grayImage = rgb2gray(Image);
imshow(grayImage);
% display(grayImage);
% Get the dimensions of the image.  numberOfColorBands
should be = 1.
[rows, columns] = size(grayImage);

tic
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Create Patches Image -- Construction stage

% block size and that all blocks will be the same size.
blockSizeR = 35; % Rows in block.
blockSizeC = 35; % Columns in block.
horSlidingStep = 20;
verSlidingStep = 20;

numColPatches = floor((columns - blockSizeC) /
horSlidingStep) + 1 ;
numRowPatches = floor((rows - blockSizeR) /
verSlidingStep) + 1 ;
%yIdxs = [1:patchSz(1):imSz(1) imSz(1)+1];
%patches = cell(numRowPatches,numColPatches);

%calculate size of Patched Image
numColPatchedImage = blockSizeR*blockSizeC ;
numRowPatchedImage = numRowPatches*numColPatches;
Dtransp = zeros(numRowPatchedImage,
numColPatchedImage);

% part of image will not be proccess due to image
size an sliding step
% size. So cut original image to new size
new_rows = floor(rows / verSlidingStep) *
verSlidingStep;
new_columns = floor(columns / horSlidingStep) *
horSlidingStep;

```

```
grayImage = grayImage(1:new_rows, 1:new_columns);  
  
% e.g. [1 9; 5 0] => indexes are [1 2 3 4]  
% we will save indexes from original image to  
linearIndexesMap  
linearIndexes = reshape(1:numel(grayImage), [new_rows,  
new_columns]);  
linearIndexesMap = Dtransp;  
  
coordR = 1;  
iter = 1;  
for i = 1:numRowPatches  
    coordC = 1;  
    for j = 1:numColPatches  
        %patches{i,j} =  
grayImage(coordR:coordR+blockSizeR - 1 ,  
coordC:coordC+blockSizeC - 1);  
        % patch is block, e.g. 50x50  
patch = grayImage(coordR:coordR+blockSizeR - 1  
, coordC:coordC+blockSizeC - 1);  
vectorizedPatch = reshape(patch,1,[]);  
Dtransp(iter,:) = vectorizedPatch;  
  
        % fill one line of linearIndexesMap, e.g. 2500  
indexes  
patchLinearIndexes =  
linearIndexes(coordR:coordR+blockSizeR - 1 ,  
coordC:coordC+blockSizeC - 1);  
vectorizedPatchLinInd =  
reshape(patchLinearIndexes,1,[]);  
linearIndexesMap(iter, :) =  
vectorizedPatchLinInd;  
  
        %display(patches{i,j});  
%imshow(patches{i,j});  
coordC = coordC + horSlidingStep;  
iter = iter + 1;  
end  
  
coordR = coordR + verSlidingStep;  
end  
  
%%%%%%%%%%%%%%  
%%  
% Realise Algorithm 1
```

```
% TODO: check if we can process without transponation
D = Dtransp.';
```

```
% image model:                                fD(x,y) = fT(x,y) +
fB(x,y) + fN(x,y)
% Infrared Patch-image (IPI) model: D = B + T + N
```

Removed

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Reconstruction image with 1D median filter
% - run over all orig image indexes
% - run over all patched Image(Tcur)
% - collect values with the same indexes to vector
% - get median 1D filter value from vector
% - set value to new original image
```

```
Ttransp = Tcur.';
```

```
max_indx = new_rows * new_columns;
% new linear image
linear_fT = zeros(1, max_indx);
% linear size of the patched image
linearSizePatchedImage = numRowsPatchedImage *
numColPatchedImage;
```

```
% -----
% -----
```

```
% not optimizes solution
% for indx = 1: max_indx
%
%     pixel_values = zeros(100,1); % TODO calculate
100 value
%     indx_px = 1;
%     for m = 1:numRowPatchedImage
%         for n = 1:numColPatchedImage
%             if linearIndexesMap(m,n) == indx
%                 pixel_values(indx_px) = Ttransp(m,n);
%                 indx_px = indx_px + 1;
%             end
%         end
%     end
%
%     %Shrink array
```

```

%      pixel_values = pixel_values(1:indx_px-1);
%      linear_fT(indx) = median(pixel_values);
% end
%-----
-----

proc_time = toc;
convergence
k
display(proc_time);
% show target image
imshow(fT);

%TODO: Get Light and Black Objects using threashold
%TODO: Draw Strobes
%
```

ДОДАТОК Б

Програма алгоритму оцінювання параметрів руху цілі з виявленням маневру

```

clear all
close all
MonteCarlo = 1;

ks=3;% 3сигма

ps=0;
kso=20;%коэф для 0.5 обл ЛО по ско измерений
ksor=40;

Pd=1;
%СКО ошибки измерения в pixels
px = 0.3; % 1 / sqrt(12)
py = 0.3;

N = 6; %количество параметров
NumHip = 3; %количество гипотез

% verLT=0.001;
% lambda = verLT/(3.465^3*pr*pb*pe*rg^2);
lambda=0;
lambda=0.025;
%lambda=0.01;
% lambda = 0,05 --> 0.05 * 1280* 720 = 46080 ложиков на
изображении в среднем
% строб размером 20 будет иметь в среднем 20 * 0.05 = 1 ложную
отметку

load TRAEK
NN = length(xtr); % Длина трека
Length=NN-2; % Длина фильтрации
T,
delay=1/25/5; % задержка вывода

%720p (1280?720 пікселів)
xx(1)=0; xx(2)=1280; % granici grafika
yy(1)=0; yy(2)=720; % granici grafika

lsx=zeros(1,200);%koord lognogo izmerernia po x
lsy=zeros(1,200);%koord lognogo izmerernia po y

% pixel/sec^2
A1=1;
A2=30;
A3=60;
A=1; %
vA1 = A1/T; vA2 = A2/T; vA3 = A3/T;
%матрица перехода

```

```

F(:, :, 1) = [ 1 T 0      0 0 0;
               0 1 0      0 0 0;
               0 0 0      0 0 0;
               0 0 0      1 T 0;
               0 0 0      0 1 0;
               0 0 0      0 0 0];

F(:, :, 2) = [ 1 T T^2/2 0 0 0;
               0 1 T      0 0 0;
               0 0 1      0 0 0;
               0 0 0      1 T T^2/2;
               0 0 0      0 1 T;
               0 0 0      0 0 1];
F(:, :, 3) = F(:, :, 2);

H = [1 0 0 0 0 0;
      0 0 0 1 0 0];

ppp=0.9;
PP=[   ppp          0.5*(1-ppp)      0.5*(1-ppp);
      0.5*(1-ppp)      ppp          0.5*(1-ppp);
      0.5*(1-ppp)      0.5*(1-ppp)      ppp      ];
%матрица вероятностей переходов

%матожидание и дисперсия оптимального положения
stekm=zeros(1,Length);
nepm=zeros(1,Length);
psm=zeros(1,Length);
xom=zeros(1,Length);
xod=zeros(1,Length);
yom=zeros(1,Length);
yod=zeros(1,Length);

%матожидание и дисперсия предсказаного положения
xzm=zeros(1,Length);
xzd=zeros(1,Length);
yzm=zeros(1,Length);
yzd=zeros(1,Length);

%матожидание и дисперсия оптимальной скорости
Vxom=zeros(1,Length);
Vxod=zeros(1,Length);
Vyom=zeros(1,Length);
Vyod=zeros(1,Length);

%матожидание и дисперсия измеренного положения
xim=zeros(1,Length);
xid=zeros(1,Length);
yim=zeros(1,Length);
yid=zeros(1,Length);

xomK = zeros(1,Length);
yomK = zeros(1,Length);

```

```

xzmK = zeros(1,Length);
yzmK = zeros(1,Length);

VxomK = zeros(1,Length);
VyomK = zeros(1,Length);

pRxK = zeros(1,Length);
pRyK = zeros(1,Length);

opt = zeros(1,Length);
suma_weroyat = zeros(1,Length);

Pobn1 = zeros(1,Length);
Pobn2 = zeros(1,Length);

podx = zeros(1,Length);
RealHip = zeros(1,Length);

for j0 = 1 : NumHip           % ускорение модели
    if (1 == j0)      vA = vA1;
    elseif (2 == j0)  vA = vA2;
    elseif (3 == j0)  vA = vA3;
    end

    G(:, :, j0) = [vA*T^3/6      0;
                   vA*T^2/2      0;
                   vA*T          0;
                   0             vA*T^3/6;
                   0             vA*T^2/2;
                   0             vA*T      ];

    Q(:, :, j0) = G(:, :, j0) * G(:, :, j0)';
end
% G, pause

% Check <<<<<
Rx=px^2;
Ry=py^2;
Rxy=0;

R = [ Rx      Rxy; %
      Rxy     Ry   ];

for i = 1 : MonteCarlo %NACHALNIE
USLOVIY!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
    % Rasmatrivaem 2 takta dlya opredeleniya skorosti celi
    U0 = [xtr(1); vxtr(1); ytr(1); vytr(1)];
%начальное положение и скорость
    % check <<<<<
    xil=round(xtr(1));

```



```

yil=round(ytr(1));

U = [xtr(2); vxtr(2); ytr(2); vytr(2)];

xi=round(xtr(2));
yi=round(ytr(2));

Po(1,1)=Rx;          Po(1,2)=Rx/T;          Po(1,3)=0;
Po(1,4)=Rxy;         Po(1,5)=Rxy/T;          Po(1,6)=0;
Po(2,1)=Po(1,2);     Po(2,2)=2*Rx/T^2+A^2*T^2; Po(2,3)=0;
Po(2,4)=Rxy/T;       Po(2,5)=2*Rxy/T^2;      Po(2,6)=0;
Po(3,1)=0;           Po(3,2)=0;
Po(3,3)=A^2; Po(3,4)=0; Po(3,5)=0;
Po(3,6)=0;
Po(4,1)=Po(1,4);     Po(4,2)=Po(2,4);          Po(4,3)=0;
Po(4,4)=Ry;          Po(4,5)=Ry/T;          Po(4,6)=0;
Po(5,1)=Po(1,5);     Po(5,2)=Po(2,5);          Po(5,3)=0;
Po(5,4)=Po(4,5); Po(5,5)=2*Ry/T^2+A^2*T^2; Po(5,6)=0;
Po(6,1)=0;           Po(6,2)=0;          Po(6,3)=0;
Po(6,4)=0;           Po(6,5)=0;          Po(6,6)=A^2;

for j0 = 1:NumHip
    if (1 == j0) A = A1;
    elseif (2 == j0) A = A2;
    elseif (3 == j0) A = A3;
    end

    x(:,j0) = [xi, (xi-xil)/T, 0, yi, (yi-yil)/T, 0];
    % начальное оптимальное значение параметров
    Po(2,2)=2*Rx/T^2+A^2*T^2; Po(3,3)=A^2;
    Po(5,5)=2*Ry/T^2+A^2*T^2; Po(6,6)=A^2;
    P(:, :, j0) = zeros(N,N);
    P(:, :, j0) = Po;
    % начальная оптимальная матрица ошибок оценки
    W(j0) = 1 / NumHip;
end

for step = 1:Length %ТРАЕКТОРИЯ!!!!!!!!!!!!!!!!!!!!!!
    filtraciya s 3-go shaga
        %%%%%%%%%%% Prognoz na 3+ shag
        %%%%%%%%%%%5

```

Removed